# 5과목 정보시스템 구축 관리

## 핵심 302 구조적 방법론

구조적 방법론은 정형화된 분석 절차에 따라 사용자 요구사항을 파악하여 문서화하는 처리(Precess) 중심의 방법론이다.

- 1960년대까지 가장 많이 적용되었던 소프트웨어 개발 방법론이다.
- 쉬운 이해 및 검증이 가능한 프로그램 코드를 생성하는 것이 목적이다.
- 복잡한 문제를 다루기 위해 분할과 정복(Divide and Conquer) 원리를 적용한다.
- 구조적 방법론의 절차

타당성 검토 단계 → 계획 단계 → 요구사항 단계 → 설계 단계 → 구현 단계 → 시험 단계 → 운용/유지보수 단계

## 핵심 303 정보공학 방법론

정보공학 방법론은 정보 시스템의 개발을 위해 계획, 분석, 설계, 구축에 정형화된 기법들을 상호 연관성 있게 통합 및 적용하는 자료(Data) 중심의 방법론이다.

- 정보 시스템 개발 주기를 이용하여 대규모 정보 시스템을 구축하는데 적합하다.
- 정보공학 방법론의 절차

정보 전략 계획 수립 단계 → 업무 영역 분석 단계 → 업무 시스템 설계 단계 → 업무 시스템 구축 단계

## 핵심 304 객체지향 방법론

객체지향 방법론은 현실 세계의 개체(Entity)를 기계의 부품처럼 하나의 객체(Object)로 만들어, 소프트웨어를 개발할 때 기계의 부품을 조립하듯이 객체들을 조립해서 필요한 소프트웨어를 구현하는 방법론이다.

- 객체지향 방법론은 구조적 기법의 문제점으로 인한 소프트웨어 위기의 해결책으로 채택되었다.
- 객체지향 방법론의 구성 요소에는 객체(Object), 클래스(Class), 메시지(Message) 등이 있다.
- 객체지향 방법론의 기본 원칙에는 캡슐화(Encapsulation), 정보 은닉(Information Hiding), 추상화(Abstraction), 상속성(Inheritance), 다형성(Polymorphism) 등이 있다.
- 객체지향 방법론의 절차

요구 분석 단계 → 설계 단계 → 구현 단계 → 테스트 및 검증 단계 → 인도 단계

## 핵심 305 컴포넌트 기반 방법론

컴포넌트 기반(CBD; Component Based Design) 방법론은 기존의 시스템이나 소프트웨어를 구성하는 컴포넌트를 조합하여 하나의 새로운 애플리케이션을 만드는 방법론이다.

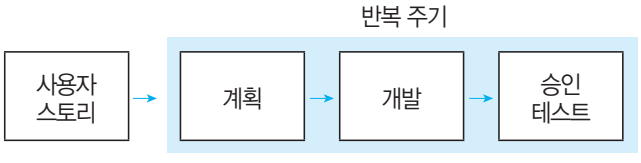
- 컴포넌트의 재사용(Reusability)이 가능하여 시간과 노력을 절감할 수 있다.
- 새로운 기능을 추가하는 것이 간단하여 확장성이 보장된다.
- 유지 보수 비용을 최소화하고 생산성 및 품질을 향상시킬 수 있다.
- 컴포넌트 기반 방법론의 절차

개발 준비 단계 → 분석 단계 → 설계 단계 → 구현 단계 → 테스트 단계 → 전개 단계 → 인도 단계

## 핵심 306 애자일 방법론

애자일(Agile)은 '민첩한', '기민한'이라는 의미로, 애자일 방법론은 고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발 과정을 진행하는 방법론이다.

- 소규모 프로젝트, 고도로 숙달된 개발자, 급변하는 요구사항에 적합하다.
- 애자일 방법론의 대표적인 종류에는 익스트림 프로그래밍(XP; eXtreme Programming), 스크럼(Scrum), 칸반(Kanban), 크리스탈(Crytal) 등이 있다.
- 애자일 방법론의 절차



## 핵심 307 소프트웨어 재사용의 개요

소프트웨어 재사용(Software Reuse)은 이미 개발되어 인정받은 소프트웨어의 전체 혹은 일부분을 다른 소프트웨어 개발이나 유지에 사용하는 것이다.

- 소프트웨어 개발의 품질과 생산성을 높이기 위한 방법으로, 기존에 개발된 소프트웨어와 경험, 지식 등을 새로운 소프트웨어에 적용한다.
- 재사용의 이점
  - 개발 시간과 비용을 단축시킨다.
  - 소프트웨어 품질을 향상시킨다.
  - 소프트웨어 개발의 생산성을 향상시킨다.
  - 프로젝트 실패의 위험을 감소시킨다.
  - 시스템 구축 방법에 대한 지식을 공유하게 된다.
  - 시스템 명세, 설계, 코드 등 문서를 공유하게 된다.

## 핵심 308 소프트웨어 재사용 방법

소프트웨어 재사용 방법에는 합성 중심 방법과 생성 중심 방법이 있다.

| | |
|---|---|
| 합성 중심 (Composition-Based) | 전자 칩과 같은 소프트웨어 부품, 즉 블록(모듈)을 만들어서 끼워 맞추어 소프트웨어를 완성시키는 방법으로, 블록 구성 방법이라고도 한다. |
| 생성 중심 (Generation-Based) | 추상화 형태로 쓰여진 명세를 구체화하여 프로그램을 만드는 방법으로, 패턴 구성 방법이라고도 한다. |

## 핵심 309 소프트웨어 재공학의 개요

소프트웨어 재공학(Software Reengineering)은 새로운 요구에 맞도록 기존 시스템 을 이용하여 보다 나은 시스템을 구축하고, 새로운 기능을 추가하여 소프트웨어 성능을 향상시키는 것이다.

- 유지보수 비용이 소프트웨어 개발 비용의 대부분을 차지하는 문제를 염두에 두어 기존 소프트웨어의 데이터와 기능들의 개조 및 개선을 통해 유지보수성과 품질을 향상 시키려는 기술이다.
- 유지보수 생산성 향상을 통해 소프트웨어 위기를 해결하는 방법이다.
- 기존 소프트웨어의 기능을 개조하거나 개선하므로, 예방(Preventive) 유지보수 측면에서 소프트웨어 위기를 해결하는 방법이라고 할 수 있다.
- 소프트웨어 재공학도 자동화된 도구를 사용하여 소프트웨어를 분석하고 수정하는 과정을 포함한다.
- 소프트웨어의 수명이 연장되고, 소프트웨어 기술이 향상될 뿐만 아니라 소프트웨어의 개발 기간도 단축된다.
- 소프트웨어에서 발생할 수 있는 오류가 줄어들고, 비용이 절감된다.



## 핵심 310 CASE의 개요

CASE(Computer Aided Software Engineering)는 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정 전체 또는 일부를 컴퓨터와 전용 소프트웨어 도구를 사용하여 자동화하는 것이다.

- 객체지향 시스템, 구조적 시스템 등 다양한 시스템에서 활용되는 자동화 도구(CASE Tool)이다.
- 소프트웨어, 하드웨어, 데이터베이스, 테스트 등을 통합하여 소프트웨어를 개발하는 환경을 조성한다.
- 소프트웨어 생명 주기의 전체 단계를 연결해 주고 자동화해 주는 통합된 도구를 제공해 주는 기술이다.
- 소프트웨어 개발 도구와 방법론이 결합된 것으로, 정형화된 구조 및 방법(메커니즘)을 소프트웨어 개발에 적용하여 생산성 향상을 구현하는 공학 기법이다.
- 소프트웨어 개발의 모든 단계에 걸쳐 일관된 방법론을 제공하는 자동화 도구들을 지원하고, 개발자들은 이 도구를 사용하여 소프트웨어 개발의 표준화를 지향하며, 자동화의 이점을 얻을 수 있게 해준다.
- CASE의 주요 기능 : 소프트웨어 생명 주기 전 단계의 연결, 다양한 소프트웨어 개발 모형 지원, 그래픽 지원 등

## 핵심 311 소프트웨어 비용 산정

소프트웨어 비용 산정은 소프트웨어의 개발 규모를 소요되는 인원, 자원, 기간 등으로 확인하여 실행 가능한 계획을 수립하기 위해 필요한 비용을 산정하는 것이다.

- 소프트웨어 비용 산정을 너무 높게 산정할 경우 예산 낭비와 일의 효율성 저하를 초래할 수 있고, 너무 낮게 산정할 경우 개발자의 부담이 가중되고 품질문제가 발생할 수 있다.
- 소프트웨어 비용 산정 기법에는 하향식 비용 산정 기법과 상향식 비용 산정 기법이 있다.
- 소프트웨어 비용을 결정하는 요소에는 프로젝트 요소, 자원 요소, 생산성 요소가 있다.

| | |
|---|---|
| **프로젝트 요소** | 제품 복잡도, 시스템 크기, 요구되는 신뢰도 |
| **자원 요소** | 인적 자원, 하드웨어 자원, 소프트웨어 자원 |
| **생산성 요소** | 개발자 능력, 개발 기간 |

## 핵심 312 하향식 비용 산정 기법

하향식 비용 산정 기법은 과거의 유사한 경험을 바탕으로 전문 지식이 많은 개발자들이 참여한 회의를 통해 비용을 산정하는 비과학적인 방법이다.

| | |
|---|---|
| **전문가 감정 기법** | • 조직 내에 있는 경험이 많은 두 명 이상의 전문가에게 비용 산정을 의뢰하는 기법이다.<br>• 가장 편리하고 신속하게 비용을 산정할 수 있으며, 의뢰자로부터 믿음을 얻을 수 있다.<br>• 새로운 프로젝트에는 과거의 프로젝트와 다른 요소들이 있다는 것을 간과할 수 있다.<br>• 새로운 프로젝트와 유사한 프로젝트에 대한 경험이 없을 수 있다.<br>• 개인적이고 주관적일 수 있다. |
| **델파이 기법** | • 전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 기법이다.<br>• 전문가들의 편견이나 분위기에 지배되지 않도록 한 명의 조정자와 여러 전문가로 구성된다. |

- 비용 산정 순서

❶ 조정자는 각 비용 산정 요원에게 시스템 정의서와 산정한 비용 내역을 기록할 서식을 제공한다.

❷ 산정 요원들은 정의서를 분석하여 익명으로 그들 나름대로의 비용을 산정한다.

❸ 조정자는 산정 요원들의 반응을 요약하여 배포한다.

❹ 산정 요원들은 이전에 산정한 결과를 이용하여 다시 익명으로 산정한다.

❺ 요원들 간의 의견이 거의 일치할 때까지 이 과정을 반복한다.

## 핵심 313 상향식 비용 산정 기법

상향식 비용 산정 기법은 프로젝트의 세부적인 작업 단위별로 비용을 산정한 후 집계하여 전체 비용을 산정하는 방법이다.

### LOC(원시 코드 라인 수, source Line Of Code) 기법

LOC 기법은 소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법이다.

- 측정이 용이하고 이해하기 쉬워 가장 많이 사용된다.
- 예측치를 이용하여 생산성, 노력, 개발 기간 등의 비용을 산정한다.

$$\text{예측치} = \frac{a+4m+b}{6}$$ 단, a : 낙관치, b : 비관치, m : 기대치(중간치)

- 산정 공식
  - 노력(인월) = 개발 기간 × 투입 인원
    = LOC / 1인당 월평균 생산 코드 라인 수
  - 개발 비용 = 노력(인월) × 단위 비용(1인당 월평균 인건비)
  - 개발 기간 = 노력(인월) / 투입 인원
  - 생산성 = LOC / 노력(인월)

### 개발 단계별 인월수(Effort Per Task) 기법

개발 단계별 인월수 기법은 LOC 기법을 보완하기 위한 기법으로, 각 기능을 구현시키는 데 필요한 노력을 생명주기의 각 단계별로 산정한다.

- LOC 기법보다 더 정확하다.

## 핵심 314 수학적 산정 기법의 개요

수학적 산정 기법은 상향식 비용 산정 기법으로, 경험적 추정 모형, 실험적 추정 모형 이라고도 하며, 개발 비용 산정의 자동화를 목표로 한다.

- 비용을 자동으로 산정하기 위해 사용되는 공식은 과거 유사한 프로젝트를 기반으로하여 경험적으로 유도된 것이다.
- 수학적 산정 기법에는 COCOMO 모형, Putnam 모형, 기능 점수(FP) 모형 등이 있으며 각 모형에서는 지정된 공식을 사용하여 비용을 산정한다.

## 핵심 315 COCOMO의 소프트웨어 개발 유형

| | |
|---|---|
| 조직형 (Organic Mode) | • 기관 내부에서 개발된 중 · 소 규모의 소프트웨어로 일괄 자료 처리나 과학 기술 계산용, 비즈니스 자료 처리용으로 5만(50KDSI) 라인 이하의 소프트웨어를 개발하는 유형이다.<br>• 사무 처리용, 업무용, 과학용 응용 소프트웨어 개발에 적합하다. |
| 반분리형 (Semi-Detached Mode) | • 조직형과 내장형의 중간형으로 트랜잭션 처리 시스템이나 운영체제, 데이터베이스 관리 시스템 등의 30만(300KDSI) 라인 이하의 소프트웨어를 개발하는 유형이다.<br>• 컴파일러, 인터프리터와 같은 유틸리티 개발에 적합하다. |
| 내장형 (Embedded Mode) | • 초대형 규모의 트랜잭션 처리 시스템이나 운영체제 등의 30만(300KDSI)라인 이상의 소프트웨어를 개발하는 유형이다.<br>• 신호기 제어 시스템, 미사일 유도 시스템, 실시간 처리 시스템 등의 시스템 프로그램 개발에 적합하다. |

## 핵심 316 COCOMO 모형의 종류

| | |
|---|---|
| 기본(Basic)형 COCOMO | 소프트웨어의 크기(생산 코드 라인 수)와 개발 유형만을 이용하여 비용을 산정하는 모형이다. |
| 중간 (Intermediate)형 COCOMO | 기본형 COCOMO의 공식을 토대로 사용하나, 다음 4가지 특성의 15가지 요인에 의해 비용을 산정하는 모형이다.<br>• 제품의 특성 : 요구되는 신뢰도, 데이터베이스 크기, 제품의 복잡도<br>• 컴퓨터의 특성 : 수행 시간의 제한, 기억장소의 제한, 가상 기계의 안정성, Turn Around Time<br>• 개발 요원의 특성 : 분석가의 능력, 개발 분야의 경험, 가상 기계의 경험, 프로그래머의 능력, 프로그래밍 언어의 경험<br>• 프로젝트 특성 : 소프트웨어 도구의 이용, 프로젝트 개발 일정, 최신 프로그래밍 기법의 이용 |
| 발전(Detailed)형 COCOMO | 중간(Intermediate)형 COCOMO를 보완하여 만들어진 방법으로 개발 공정별로 보다 자세하고 정확하게 노력을 산출하여 비용을 산정하는 모형이다. |

## 핵심 317 Putnam 모형

Putnam 모형은 소프트웨어 생명 주기의 전 과정 동안에 사용될 노력의 분포를 가정해 주는 모형이다.

- 푸트남(Putnam)이 제안한 것으로 생명 주기 예측 모형이라고도 한다.
- 시간에 따른 함수로 표현되는 Rayleigh-Norden 곡선의 노력 분포도를 기초로 한다.
- 대형 프로젝트의 노력 분포 산정에 이용되는 기법이다.
- 개발 기간이 늘어날수록 프로젝트 적용 인원의 노력이 감소한다.

## 핵심 318 기능 점수(FP) 모형

기능 점수(Function Point) 모형은 알브레히트(Albrecht)가 제안한 것으로, 소프트웨어의 기능을 증대시키는 요인별로 가중치를 부여하고, 요인별 가중치를 합산하여 총 기능 점수를 산출하며 총 기능 점수와 영향도를 이용하여 기능 점수(FP)를 구한 후 이를 이용해서 비용을 산정하는 기법이다.

기능 점수(FP) = 총 기능 점수 × [0.65 + (0.1 × 총 영향도)]

- 소프트웨어 기능 증대 요인
  - 자료 입력(입력 양식)
  - 정보 출력(출력 보고서)
  - 명령어(사용자 질의수)
  - 데이터 파일
  - 필요한 외부 루틴과의 인터페이스

※ 자동화 추정 도구

- SLIM : Rayleigh-Norden 곡선과 Putnam 예측 모델을 기초로 하여 개발된 자동화 추정 도구
- ESTIMACS : 다양한 프로젝트와 개인별 요소를 수용하도록 FP 모형을 기초로 하여 개발된 자동화 추정 도구

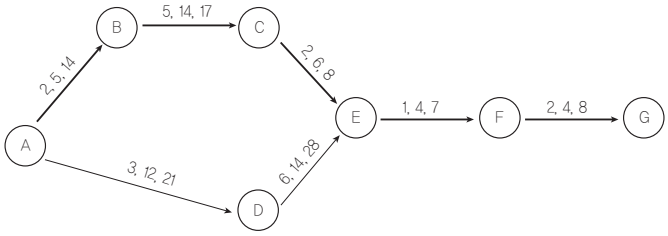
## 핵심 319 PERT

PERT(Program Evaluation and Review Technique, 프로그램 평가 및 검토 기술)는 프로젝트에 필요한 전체 작업의 상호 관계를 표시하는 네트워크로 각 작업별로 낙관적인 경우, 가능성이 있는 경우, 비관적인 경우로 나누어 각 단계별 종료 시기를 결정하는 방법이다.

- 과거에 경험이 없어서 소요 기간 예측이 어려운 소프트웨어에서 사용한다.
- 노드와 간선으로 구성되며 원 노드에는 작업을, 간선(화살표)에는 낙관치, 기대치, 비관치를 표시한다.
- 결정 경로, 작업에 대한 경계 시간, 작업 간의 상호 관련성 등을 알 수 있다.
- 다음과 같은 PERT 공식을 이용하여 작업 예측치를 계산한다.

$$\text{작업 예측치} = \frac{\text{비관치} + 4 \times \text{기대치} + \text{낙관치}}{6}$$

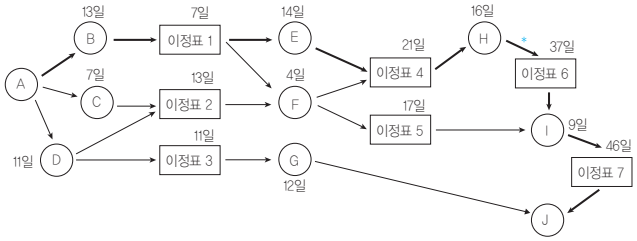
$$\text{평방 편차} = \left[\frac{(\text{비관치} - \text{낙관치})}{6}\right]^2$$




## 핵심 320 CPM

CPM(Critical Path Method, 임계 경로 기법)은 프로젝트 완성에 필요한 작업을 나열하고 작업에 필요한 소요 기간을 예측하는데 사용하는 기법이다.

- CPM은 노드와 간선으로 구성된 네트워크로 노드는 작업을, 간선은 작업 사이의 전후 의존 관계를 나타낸다.
- 원형 노드는 각 작업을 의미하며 각 작업 이름과 소요 기간을 표시하고, 박스 노드 는 이정표를 의미하며 박스 노드 위에는 예상 완료 시간을 표시한다.
- 간선을 나타내는 화살표의 흐름에 따라 각 작업이 진행되며, 전 작업이 완료된 후 다음 작업을 진행할 수 있다.
- 임계 경로는 최장 경로를 의미한다.




## 핵심 321 간트 차트

간트 차트는 프로젝트의 각 작업들이 언제 시작하고 언제 종료되는지에 대한 작업 일 정을 막대 도표를 이용하여 표시하는 프로젝트 일정표로, 시간선(Time-Line) 차트라고도 한다.

- 중간 목표 미달성 시 그 이유와 기간을 예측할 수 있게 한다.
- 사용자와의 문제점이나 예산의 초과 지출 등도 관리할 수 있게 한다.
- 자원 배치와 인원 계획에 유용하게 사용된다.
- 다양한 형태로 변경하여 사용할 수 있다.
- 작업 경로는 표시할 수 없으며, 계획의 변화에 대한 적응성이 약하다.
- 계획 수립 또는 수정 때 주관적 수치에 기울어지기 쉽다.
- 간트 차트는 이정표, 작업 일정, 작업 기간, 산출물로 구성되어 있다.
- 수평 막대의 길이는 각 작업(Task)의 기간을 나타낸다.

## 핵심 322 소프트웨어 개발 방법론 결정의 개요

소프트웨어 개발 방법론의 결정은 프로젝트 관리와 재사용 현황을 소프트웨어 개발 방법론에 반영하고, 확정된 소프트웨어 생명 주기와 개발 방법론에 맞춰 소프트웨어 개발 단계, 활동, 작업, 절차 등을 정의하는 것이다.

※ 프로젝트 관리(Project Management)

프로젝트 관리는 주어진 기간 내에 최소의 비용으로 사용자를 만족시키는 시스템을 개발하기 위한 전반적인 활동이다.

| 관리 유형 | 주요 내용 |
|---|---|
| 일정 관리 | 작업 순서, 작업 기간 산정, 일정 개발, 일정 통제 |
| 비용 관리 | 비용 산정, 비용 예산 편성, 비용 통제 |
| 인력 관리 | 프로젝트 팀 편성, 자원 산정, 프로젝트 조직 정의, 프로젝트 팀 개발, 자원 통제, 프로젝트 팀 관리 |
| 위험 관리 | 위험 식별, 위험 평가, 위험 대처, 위험 통제 |
| 품질 관리 | 품질 계획, 품질 보증 수행, 품질 통제 수행 |

## 핵심 323 ISO/IEC 12207

ISO/IEC 12207은 ISO(International Organization for Standardization, 국제표준화기구)에서 만든 표준 소프트웨어 생명 주기 프로세스로, 소프트웨어의 개발, 운영, 유지보수 등을 체계적으로 관리하기 위한 소프트웨어 생명 주기 표준을 제공한다.

- ISO/IEC 12207은 기본 생명 주기 프로세스, 지원 생명 주기 프로세스, 조직 생명 주기 프로세스로 구분한다.

| | |
|---|---|
| **기본 생명 주기 프로세스** | 획득, 공급, 개발, 운영, 유지보수 프로세스 |
| **지원 생명 주기 프로세스** | 품질 보증, 검증, 확인, 활동 검토, 감사, 문서화, 형상 관리, 문제 해결 프로세스 |
| **조직 생명 주기 프로세스** | 관리, 기반 구조, 훈련, 개선 프로세스 |

## 핵심 324 CMMI(Capability Maturity Model Integration)

CMMI(능력 성숙도 통합 모델)는 소프트웨어 개발 조직의 업무 능력 및 조직의 성숙도를 평가하는 모델로, 미국 카네기멜론 대학교의 소프트웨어 공학연구소(SEI)에서 개발하였다.

- CMMI의 소프트웨어 프로세스 성숙도는 초기, 관리, 정의, 정량적 관리, 최적화의 5단계로 구분한다.

| 단계 | 프로세스 | 특징 |
|---|---|---|
| 초기(Initial) | 정의된 프로세스 없음 | 작업자 능력에 따라 성공 여부 결정 |
| 관리(Managed) | 규칙화된 프로세스 | 특정한 프로젝트 내의 프로세스 정의 및 수행 |
| 정의(Defined) | 표준화된 프로세스 | 조직의 표준 프로세스를 활용하여 업무 수행 |
| 정량적 관리(Quantitatively Managed) | 예측 가능한 프로세스 | 프로젝트를 정량적으로 관리 및 통제 |
| 최적화(Optimizing) | 지속적 개선 프로세스 | 프로세스 역량 향상을 위해 지속적인 프로세스 개선 |

## 핵심 325 SPICE(Software Process Improvement and Capability dEtermination)

SPICE(소프트웨어 처리 개선 및 능력 평가 기준)는 정보시스템 분야에서 소프트웨어의 품질 및 생산성 향상을 위해 소프트웨어 프로세스를 평가 및 개선하는 국제 표준으로, 공식 명칭은 ISO/IEC 15504이다.

- SPICE는 5개의 프로세스 범주와 40개의 세부 프로세스로 구성된다.
- SPICE는 프로세스 수행 능력 단계를 불완전, 수행, 관리, 확립, 예측, 최적화의 6단계로 구분한다.

| 단계 | 특징 |
|---|---|
| 불완전(Incomplete) | 프로세스가 구현되지 않았거나 목적을 달성하지 못한 단계이다. |
| 수행(Performed) | 프로세스가 수행되고 목적이 달성된 단계이다. |
| 관리(Managed) | 정의된 자원의 한도 내에서 그 프로세스가 작업 산출물을 인도하는 단계이다. |

| | |
|---|---|
| 확립 (Established) | 소프트웨어 공학 원칙에 기반하여 정의된 프로세스가 수행되는 단계이다. |
| 예측 (Predictable) | 프로세스가 목적 달성을 위해 통제되고, 양적인 측정을 통해서 일관되게 수행되는 단계이다. |
| 최적화 (Optimizing) | 프로세스 수행을 최적화하고, 지속적인 개선을 통해 업무 목적을 만족시키는 단계이다. |

## 핵심 326 소프트웨어 개발 방법론 테일러링의 개요 및 고려사항

소프트웨어 개발 방법론 테일러링은 프로젝트 상황 및 특성에 맞도록 정의된 소프트웨어 개발 방법론의 절차, 사용기법 등을 수정 및 보완하는 작업이다.

- 소프트웨어 개발 방법론 테일러링 작업 시 고려해야 할 사항에는 내부적 기준과 외부적 기준이 있다.

| | |
|---|---|
| 내부적 기준 | • 목표 환경 : 시스템의 개발 환경과 유형이 서로 다른 경우 테일러링이 필요하다.<br>• 요구사항 : 프로젝트의 생명 주기 활동에서 개발, 운영, 유지보수 등 프로젝트에서 우선적으로 고려할 요구사항이 서로 다른 경우 테일러링이 필요하다.<br>• 프로젝트 규모 : 비용, 인력, 기간 등 프로젝트의 규모가 서로 다른 경우 테일러링이 필요하다.<br>• 보유 기술 : 프로세스, 개발 방법론, 산출물, 구성원의 능력 등이 서로 다른 경우 테일러링이 필요하다. |
| 외부적 기준 | • 법적 제약사항 : 프로젝트별로 적용될 IT Compliance가 서로 다른 경우 테일러링이 필요하다.<br>• 표준 품질 기준 : 금융, 제도 등 분야별 표준 품질 기준이 서로 다른 경우 테일러링이 필요하다. |

## 핵심 327 소프트웨어 개발 프레임워크

프레임워크(Framework)는 소프트웨어 개발에 공통적으로 사용되는 구성 요소와 아키텍처를 일반화하여 손쉽게 구현할 수 있도록 여러 가지 기능들을 제공해주는 반제품 형태의 소프트웨어 시스템이다.

- 선행 사업자의 기술에 의존하지 않은 표준화된 개발 기반으로 인해 사업자 종속성이 해소된다.
- 프레임워크의 주요 기능에는 예외 처리, 트랜잭션 처리, 메모리 공유, 데이터 소스 관리, 서비스 관리, 쿼리 서비스, 로깅 서비스, 사용자 인증 서비스 등이 있다.

- 프레임워크의 종류

| | |
|---|---|
| **스프링 프레임워크 (Spring Framework)** | 자바 플랫폼을 위한 오픈 소스 경량형 애플리케이션 프레임워크이다. |
| **전자정부 프레임워크** | 우리나라의 공공부문 정보화 사업 시 효율적인 정보 시스템의 구축을 지원하기 위해 필요한 기능 및 아키텍처를 제공하는 프레임워크이다. |
| **닷넷 프레임워크 (.NET Framework)** | Windows 프로그램의 개발 및 실행 환경을 제공하는 프레임워크로, Microsoft 사에서 통합 인터넷 전략을 위해 개발하였다. |

## 핵심 328 프레임워크의 특성

| | |
|---|---|
| **모듈화 (Modularity)** | • 프레임워크는 캡슐화를 통해 모듈화를 강화하고 설계 및 구현의 변경에 따른 영향을 최소화함으로써 소프트웨어의 품질을 향상시킨다.<br>• 프레임워크는 개발표준에 의한 모듈화로 인해 유지 보수가 용이하다. |
| **재사용성 (Reusability)** | 프레임워크는 재사용 가능한 모듈들을 제공함으로써 예산 절감, 생산성 향상, 품질 보증이 가능하다. |
| **확장성 (Extensibility)** | 프레임워크는 다형성(Polymorphism)을 통한 인터페이스 확장이 가능하여 다양한 형태와 기능을 가진 애플리케이션 개발이 가능하다. |
| **제어의 역흐름 (Inversion of Control)** | 개발자가 관리하고 통제해야 하는 객체들의 제어를 프레임워크에 넘김으로써 생산성을 향상시킨다. |

## 핵심 329 네트워크 관련 신기술

| | |
|---|---|
| **IoT(Internet of Things, 사물 인터넷)** | 정보 통신 기술을 기반으로 실세계(Physical World)와 가상 세계(Virtual World)의 다양한 사물들을 인터넷으로 서로 연결하여 진보된 서비스를 제공하기 위한 서비스 기반 기술이다. |
| **M2M(Machine to Machine, 사물 통신)** | • 무선 통신을 이용한 기계와 기계 사이의 통신이다.<br>• M2M은 변압기 원격 감시, 전기, 가스 등의 원격 검침, 무선 신용카드 조회기, 무선 보안단말기, 버스 운행 시스템, 위치 추적 시스템, 시설물 관리 등을 무선으로 통합하여 상호 작용하는 통신이다. |
| **모바일 컴퓨팅 (Mobile Computing)** | 휴대형 기기로 이동하면서 자유로이 네트워크에 접속하여 업무를 처리할 수 있는 환경을 말한다. |

| | |
|---|---|
| **클라우드컴퓨팅 (Cloud Computing)** | • 각종 컴퓨팅 자원을 중앙 컴퓨터에 두고 인터넷 기능을 갖는 단말기로 언제 어디서나 인터넷을 통해 컴퓨터 작업을 수행할 수 있는 환경을 의미한다.<br>• 중앙 컴퓨터는 복수의 데이터 센터를 가상화 기술로 통합한 대형 데이터 센터로, 각종 소프트웨어, 데이터, 보안 솔루션 기능 등 컴퓨팅 자원을 보유하고 있다. |
| **모바일 클라우드 컴퓨팅 (MCC; Mobile Cloud Computing)** | 클라우드 서비스를 이용하여 소비자와 소비자의 파트너가 모바일 기기로 클라우드 컴퓨팅 인프라를 구성하여 여러 가지 정보와 자원을 공유하는 ICT(Information and Communications Technologies) 기술을 의미한다. |
| **인터클라우드 컴퓨팅 (Inter-Cloud Computing)** | 각기 다른 클라우드 서비스를 연동하거나 컴퓨팅 자원의 동적 할당이 가능하도록 여러 클라우드 서비스 제공자들이 제공하는 클라우드 서비스나 자원을 연결하는 기술을 말한다. |
| **메시 네트워크 (Mesh Network)** | 차세대 이동통신, 홈네트워킹, 공공 안전 등 특수 목적을 위한 새로운 방식의 네트워크 기술로, 대규모 디바이스의 네트워크 생성에 최적화되어 있다. |
| **와이선(Wi-SUN)** | 스마트 그리드와 같은 장거리 무선 통신을 필요로 하는 사물 인터넷(IoT) 서비스를 위한 저전력 장거리(LPWA; Low-Power Wide Area) 통신 기술이다. |
| **NDN (Named Data Networking)** | 콘텐츠 자체의 정보와 라우터 기능만으로 데이터 전송을 수행하는 기술로, 클라이언트와 서버가 패킷의 헤더에 내장되어 있는 주소 정보를 이용하여 연결되던 기존의 IP(Internet Protocol) 망을 대체할 새로운 인터넷 아키텍처로 떠오르고 있다. |
| **NGN(Next Generation Network, 차세대 통신망)** | ITU-T에서 개발하고 있는 유선망 기반의 차세대 통신망으로, 유선망뿐만 아니라 이동 사용자를 목표로 하며, 이동통신에서 제공하는 완전한 이동성(Full Mobility) 제공을 목표로 개발되고 있다. |
| **SDN(Software Defined Networking, 소프트웨어 정의 네트워킹)** | 네트워크를 컴퓨터처럼 모델링하여 여러 사용자가 각각의 소프트웨어들로 네트워킹을 가상화하여 제어하고 관리하는 네트워크이다. |
| **NFC (Near Field Communication, 근거리 무선 통신)** | • 고주파(HF)를 이용한 근거리 무선 통신 기술이다.<br>• NFC는 Ecma 340, ISO/IEC 18092 표준으로, 아주 가까운 거리에서 양방향 통신을 지원하는 RFID 기술의 일종이다. |
| **UWB(Ultra WideBand, 초광대역)** | 짧은 거리에서 많은 양의 디지털 데이터를 낮은 전력으로 전송하기 위한 무선 기술로 무선 디지털 펄스라고도 하며, 블루투스와 비교되는 기술이다. |
| **피코넷 (PICONET)** | 여러 개의 독립된 통신장치가 블루투스 기술이나 UWB 통신 기술을 사용하여 통신망을 형성하는 무선 네트워크 기술이다. |
| **WBAN (Wireless Body Area Network)** | 웨어러블(Wearable) 또는 몸에 심는(Implant) 형태의 센서나 기기를 무선으로 연결하는 개인 영역 네트워킹 기술이다. |
| **GIS(Geographic Information System, 지리정보 시스템)** | 지리적인 자료를 수집 · 저장 · 분석 · 출력할 수 있는 컴퓨터 응용 시스템으로, 위성을 이용해 모든 사물의 위치 정보를 제공해 주는 것을 말한다. |
| **USN(Ubiquitous Sensor Network, 유비쿼터스 센서 네트워크)** | 각종 센서로 수집한 정보를 무선으로 수집할 수 있도록 구성한 네트워크를 말한다. 즉 필요한 모든 것(곳)에 RFID 태그를 부착하고, 이를 통하여 사물의 인식 정보는 물론 주변의 환경정보까지 탐지하여 이를 네트워크에 연결하여 정보를 관리하는 것을 의미한다. |
| **SON(Self Organizing Network, 자동 구성 네트워크)** | • 주변 상황에 맞추어 스스로 망을 구성하는 네트워크를 말한다.<br>• SON의 목적은 통신망 커버리지 및 전송 용량 확장의 경제성 문제를 해결하고, 망의 운영과 관리의 효율성을 높이는 것이다. |
| **애드 혹 네트워크 (Ad-hoc Network)** | 재난 현장과 같이 별도의 고정된 유선망을 구축할 수 없는 장소에서 모바일 호스트(Mobile Host)만을 이용하여 구성한 네트워크로, 망을 구성한 후 단기간 사용되는 경우나 유선망을 구성하기 어려운 경우에 적합하다. |
| **네트워크 슬라이싱 (Network Slicing)** | 3GPP를 포함한 여러 글로벌 이동통신 표준화 단체가 선정한 5G(IMT-2020)의 핵심기술 중 하나로, 네트워크에서 하나의 물리적인 코어 네트워크 인프라(Infrastructure)를 독립된 다수의 가상 네트워크로 분리하여 각각의 네트워크를 통해 다양한 고객 맞춤형 서비스를 제공하는 것을 목적으로 하는 네트워크 기술이다. |
| **저전력 블루투스 기술 (BLE; Bluetooth Low Energy)** | 일반 블루투스와 동일한 2.4GHz 주파수 대역을 사용하지만 연결되지 않은 대기 상태에서는 절전 모드를 유지하는 기술이다. |
| **지능형 초연결망** | 과학기술정보통신부 주관으로 추진 중인 사업으로, 스마트 시티, 스마트 스테이션 등 4차 산업혁명 시대를 맞아 새로운 변화에 따라 급격하게 증가하는 데이터 트래픽을 효과적으로 수용하기 위해 시행되는 정부 주관 사업이다. |
| **파장 분할 다중화(WDM, Wavelength Division Multiplexing)** | • 광섬유를 이용한 통신 기술의 하나로, 파장이 서로 다른 복수의 신호를 보냄으로써 여러 대의 단말기가 동시에 통신 회선을 사용할 수 있도록 하는 것이다.<br>• 파장이 다른 광선끼리는 서로 간섭을 일으키지 않는 성질을 이용한 기술이다. |


| | |
|---|---|
| 소프트웨어 정의 데이터 센터 (SDDC, Software Defined Data Center) | 데이터 센터의 모든 자원을 가상화하여 인력의 개입 없이 소프트웨어 조작만으로 관리 및 제어되는 데이터 센터를 의미한다. |
| 개방형 링크드 데이터 (LOD, Linked Open Data) | Linked Data와 Open Data의 합성어로, 누구나 사용할 수 있도록 웹상에 공개된 연계 데이터를 의미한다. |

## 핵심 330 네트워크(Network) 설치 구조

| | |
|---|---|
| 성형 (Star, 중앙 집중형) | • 중앙에 중앙 컴퓨터가 있고, 이를 중심으로 단말장치들이 연결되는 중앙 집중식의 네트워크 구성 형태이다.<br>• 포인트 투 포인트(Point-to-Point) 방식으로 회선을 연결한다. |
| 링형 (Ring, 루프형) | • 컴퓨터와 단말장치들을 서로 이웃하는 것끼리 포인트 투 포인트(Point-to-Point) 방식으로 연결시킨 형태이다.<br>• 분산 및 집중 제어 모두 가능하다.<br>• 데이터는 단방향 또는 양방향으로 전송할 수 있으며, 단방향 링의 경우 컴퓨터, 단말장치, 통신 회선 중 어느 하나라도 고장나면 전체 통신망에 영향을 미친다. |
| 버스형(Bus) | • 한 개의 통신 회선에 여러 대의 단말장치가 연결되어 있는 형태이다.<br>• 물리적 구조가 간단하고, 단말장치의 추가와 제거가 용이하다.<br>• 단말장치가 고장나더라도 통신망 전체에 영향을 주지 않기 때문에 신뢰성을 높일 수 있다. |
| 계층형 (Tree, 분산형) | 중앙 컴퓨터와 일정 지역의 단말장치까지는 하나의 통신 회선으로 연결시키고, 이웃하는 단말장치는 일정 지역 내에 설치된 중간 단말장치로부터 다시 연결시키는 형태이다. |
| 망형(Mesh) | • 모든 지점의 컴퓨터와 단말장치를 서로 연결한 형태로, 노드의 연결성이 높다.<br>• 많은 단말장치로부터 많은 양의 통신을 필요로 하는 경우에 유리하다.<br>• 보통 공중 데이터 통신망에서 사용되며, 통신 회선의 총 경로가 가장 길다.<br>• 모든 노드를 망형으로 연결하려면 노드의 수가 n개일 때, n(n-1)/2개의 회선이 필요하고 노드당 n-1개의 포트가 필요하다. |

## 핵심 331 네트워크 분류

네트워크는 각 사이트들이 분포되어 있는 지리적 범위에 따라 LAN과 WAN으로 분류된다.

| | |
|---|---|
| 근거리 통신망 (LAN; Local Area Network) | • 회사, 학교, 연구소 등에서 비교적 가까운 거리에 있는 컴퓨터, 프린터, 테이프 등과 같은 자원을 연결하여 구성한다.<br>• 주로 자원 공유를 목적으로 사용한다.<br>• 사이트 간의 거리가 짧아 데이터의 전송 속도가 빠르고, 에러 발생율이 낮다.<br>• 근거리 통신망에서는 주로 버스형이나 링형 구조를 사용한다. |
| 광대역 통신망 (WAN; Wide Area Network) | • 국가와 국가 혹은 대륙과 대륙 등과 같이 멀리 떨어진 사이트들을 연결하여 구성한다.<br>• 사이트 간의 거리가 멀기 때문에 통신 속도가 느리고, 에러 발생률이 높다.<br>• 일정한 지역에 있는 사이트들을 근거리 통신망으로 연결한 후 각 근거리 통신망을 연결하는 방식을 사용한다. |

## 핵심 332 LAN의 표준안

### IEEE 802의 주요 표준 규격

IEEE 802 위원회에서 지정한 LAN의 표준 규격은 다음과 같다.

| 표준 규격 | 내용 |
|---|---|
| 802.1 | 전체의 구성, OSI 참조 모델과의 관계, 통신망 관리 등에 관한 규약이다. |
| 802.2 | 논리 링크 제어(LLC) 계층에 관한 규약이다. |
| 802.3 | CSMA/CD 방식의 매체 접근 제어 계층에 관한 규약이다. |
| 802.4 | 토큰 버스 방식의 매체 접근 제어 계층에 관한 규약이다. |
| 802.5 | 토큰 링 방식의 매체 접근 제어 계층에 관한 규약이다. |
| 802.6 | 도시형 통신망(MAN)에 관한 규약이다. |
| 802.9 | 종합 음성/데이터 네트워크에 관한 규약이다. |
| 802.11 | 무선 LAN에 관한 규약이다. |

### 802.11의 버전

| | |
|---|---|
| 802.11 (초기 버전) | 2.4GHz 대역 전파와 CSMA/CA 기술을 사용해 최고 2Mbps까지의 전송 속도를 지원한다. |
| 802.11a | 5GHz 대역의 전파를 사용하며, OFDM 기술을 사용해 최고 54Mbps까지의 전송 속도를 지원한다. |

| | |
|---|---|
| 802.11b | 802.11 초기 버전의 개선안으로 등장하였으며, 초기 버전의 대역 전파와 기술을 사용해 최고 11Mbps의 전송 속도로 기존에 비해 5배 이상 빠르게 개선되었다. |
| 802.11e | 802.11의 부가 기능 표준으로, QoS 기능이 지원되도록 하기 위해 매체 접근 제어(MAC) 계층에 해당하는 부분을 수정하였다. |
| 802.11g | 2.4GHz 대역의 전파를 사용하지만 5GHz 대역의 전파를 사용하는 802.11a와 동일한 최고 54Mbps까지의 전송 속도를 지원한다. |
| 802.11n | 2.4GHz 대역과 5GHz 대역을 사용하는 규격으로, 최고 600Mbps까지의 전송 속도를 지원한다. |

## 핵심 333 스위치

### 스위치(Switch) 분류

스위치는 브리지와 같이 LAN과 LAN을 연결하여 훨씬 더 큰 LAN을 만드는 장치로, OSI 7 계층의 Layer에 따라 L2, L3, L4, L7으로 분류된다.

| | |
|---|---|
| L2 스위치 | • OSI의 2계층에 속하는 장비이다.<br>• 일반적으로 부르는 스위치는 L2 스위치를 의미한다.<br>• MAC 주소를 기반으로 프레임을 전송한다.<br>• 동일 네트워크 간의 연결만 가능하다. |
| L3 스위치 | • OSI의 3계층에 속하는 장비이다.<br>• L2 스위치에 라우터 기능이 추가된 것으로, IP 주소를 기반으로 패킷을 전송한다.<br>• 서로 다른 네트워크 간의 연결이 가능하다. |
| L4 스위치 | • OSI 4계층에 속하는 장비이다.<br>• 로드밸런서가 달린 L3 스위치로, IP 주소 및 TCP/UDP를 기반으로 사용자들의 요구를 서버의 부하가 적은 곳에 배분하는 로드밸런싱 기능을 제공한다. |
| L7 스위치 | • OSI 7계층에 속하는 장비이다.<br>• IP 주소, TCP/UDP 포트 정보에 패킷 내용까지 참조하여 세밀하게 로드밸런싱한다. |

### 스위칭(Switch) 방식

스위치가 프레임을 전달하는 방식에 따라 Store and Forwarding, Cut-through, Fragment Free가 있다.

| | |
|---|---|
| Store and Forwarding | 데이터를 모두 받은 후 스위칭하는 방식 |
| Cut-through | 데이터의 목적지 주소만을 확인한 후 바로 스위칭하는 방식 |
| Fragment Free | Store and Forwarding과 Cut-through 방식의 장점을 결합한 방식 |

## 핵심 334 경로 제어(Routing)

경로 제어는 송 · 수신 측 간의 전송 경로 중에서 최적 패킷 교환 경로를 결정하는 기능이다.

- 최적 패킷 교환 경로란 어느 한 경로에 데이터의 양이 집중하는 것을 피하면서, 최저의 비용으로 최단 시간에 송신할 수 있는 경로를 의미한다.
- 경로 제어 요소 : 성능 기준, 경로의 결정 시간과 장소, 정보 발생지, 경로 정보의 갱신 시간
- 경로 제어 프로토콜(Routing Protocol)

| | |
|---|---|
| GP(Interior Gateway Protocol, 내부 게이트웨이 프로토콜) | • 하나의 자율 시스템(AS) 내의 라우팅에 사용되는 프로토콜이다.<br>• RIP(Routing Information Protocol)<br>– 현재 가장 널리 사용되는 라우팅 프로토콜로 거리 벡터 라우팅 프로토콜이라고도 불리며, 최단 경로 탐색에 Bellman-Ford 알고리즘이 사용된다.<br>– 소규모 동종의 네트워크(자율 시스템, AS) 내에서 효율적인 방법이다.<br>– 최대 홉(Hop) 수를 15로 제한하므로 15 이상의 경우는 도달할 수 없는 네트워크를 의미하는데 이것은 대규모 네트워크에서는 RIP를 사용할 수 없음을 의미한다.<br>• OSPF(Open Shortest Path First protocol)<br>– RIP의 단점을 해결하여 새로운 기능을 지원하는 인터넷 프로토콜로, 대규모 네트워크에서 많이 사용된다.<br>– 인터넷 망에서 이용자가 최단 경로를 선정할 수 있도록 라우팅 정보에 노드 간의 거리 정보, 링크 상태 정보를 실시간으로 반영하여 최단 경로로 라우팅을 지원한다.<br>– 최단 경로 탐색에 다익스트라(Dijkstra) 알고리즘을 사용한다.<br>– 라우팅 정보에 변화가 생길 경우 변화된 정보만 네트워크 내의 모든 라우터에 알린다.<br>– 하나의 자율 시스템(AS)에서 동작하면서 내부 라우팅 프로토콜의 그룹에 도달한다. |
| EGP(Exterior Gateway Protocol, 외부 게이트웨이 프로토콜) | 자율 시스템(AS) 간의 라우팅, 즉 게이트웨이 간의 라우팅에 사용되는 프로토콜이다. |
| BGP(Border Gateway Protocol) | • 자율 시스템(AS) 간의 라우팅 프로토콜로, EGP의 단점을 보완하기 위해 만들어졌다.<br>• 초기에 BGP 라우터들이 연결될 때에는 전체 경로 제어표(라우팅 테이블)를 교환하고, 이후에는 변화된 정보만을 교환한다. |

## 핵심 335 트래픽 제어(Traffic Control)

트래픽 제어는 네트워크의 보호, 성능 유지, 네트워크 자원의 효율적인 이용을 위해 전송되는 패킷의 흐름 또는 그 양을 조절하는 기능으로 흐름 제어, 폭주(혼합) 제어, 교착상태 방지 기법이 있다.

- **흐름 제어(Flow Control)** : 흐름 제어란 네트워크 내의 원활한 흐름을 위해 송 · 수신 측 사이에 전송되는 패킷의 양이나 속도를 규제하는 기능이다.

| | |
|---|---|
| **정지-대기 (Stop-and-Wait)** | • 수신 측의 확인 신호(ACK)를 받은 후에 다음 패킷을 전송하는 방식이다.<br>• 한 번에 하나의 패킷만을 전송할 수 있다. |
| **슬라이딩 윈도우 (Sliding Window)** | • 확인 신호, 즉 수신 통지를 이용하여 송신 데이터의 양을 조절하는 방식이다.<br>• 수신 측의 확인 신호를 받지 않더라도 미리 정해진 패킷의 수만큼 연속적으로 전송하는 방식으로, 한 번에 여러 개의 패킷을 전송할 수 있어 전송 효율이 좋다.<br>• 송신 측은 수신 측으로부터 확인 신호(ACK) 없이도 보낼 수 있는 패킷의 최대치를 미리 약속받는데, 이 패킷의 최대치가 윈도우 크기(Window Size)를 의미한다.<br>• 윈도우 크기(Window Size)는 상황에 따라 변한다. 즉, 수신 측으로부터 이전에 송신한 패킷에 대한 긍정 수신 응답(ACK)이 전달된 경우 윈도우 크기는 증가하고, 수신 측으로부터 이전에 송신한 패킷에 대한 부정 수신 응답(NAK)이 전달된 경우 윈도우 크기는 감소한다. |

- **폭주(혼잡) 제어(Congestion Control)** : 흐름 제어(Flow Control)가 송 · 수신 측 사이의 패킷 수를 제어하는 기능이라면, 폭주 제어는 네트워크 내의 패킷 수를 조절하여 네트워크의 오버플로(Overflow)를 방지하는 기능을 한다.

| | |
|---|---|
| **느린 시작 (Slow Start)** | • 윈도우의 크기를 1, 2, 4, 8, …과 같이 2배씩 지수적으로 증가시켜 초기에는 느리지만 갈수록 빨라진다.<br>• 전송 데이터의 크기가 임계 값에 도달하면 혼잡 회피 단계로 넘어간다. |
| **혼잡 회피 (Congestion Avoidance)** | 느린 시작(Slow Start)의 지수적 증가가 임계 값에 도달되면 혼잡으로 간주하고 회피를 위해 윈도우의 크기를 1씩 선형적으로 증가시켜 혼잡을 예방하는 방식이다. |

- **교착상태(Dead Lock) 방지** : 교착상태란 교환기 내에 패킷들을 축적하는 기억 공간이 꽉 차 있을 때 다음 패킷들이 기억 공간에 들어가기 위해 무한정 기다리는 현상을 말한다.

## 핵심 336 SW 관련 신기술

| | |
|---|---|
| **인공지능 (AI; Artificial Intelligence)** | 인간의 두뇌와 같이 컴퓨터 스스로 추론, 학습, 판단 등 인간지능적인 작업을 수행하는 시스템이다. |
| **뉴럴링크 (Neuralink)** | • 미국의 전기자동차 회사 테슬라(Tesla)의 CEO 일론 머스크(Elon Musk)가 사람의 뇌와 컴퓨터를 결합하는 기술을 개발하기 위해 2017년 3월 설립한 회사이다.<br>• 뉴럴링크가 개발하고 있는 기술은 '신경 레이스(Neural Lace)'로, 작은 전극을 뇌에 이식함으로써 생각을 업로드하고 다운로드하는 것을 목표로 삼고 있다. |
| **딥 러닝 (Deep Learning)** | 인간의 두뇌를 모델로 만들어진 인공 신경망(ANN; Artificial Neural Network)을 기반으로 하는 기계 학습 기술이다. |
| **전문가 시스템 (Expert System)** | 의료 진단 등과 같은 특정 분야의 전문가가 수행하는 고도의 업무를 지원하기 위한 컴퓨터 응용 프로그램이다. |
| **증강현실 (AR; Augmented Reality)** | 실제 촬영한 화면에 가상의 정보를 부가하여 보여주는 기술이다. |
| **블록체인 (Blockchain)** | P2P 네트워크를 이용하여 온라인 금융 거래 정보를 온라인 네트워크 참여자(Peer)의 디지털 장비에 분산 저장하는 기술을 의미한다. |
| **분산 원장 기술 (DLT; Distributed Ledger Technology)** | 중앙 관리자나 중앙 데이터 저장소가 존재하지 않고 P2P 망 내의 참여자들에게 모든 거래 목록이 분산 저장되어 거래가 발생할 때마다 지속적으로 갱신되는 디지털 원장을 의미한다. |
| **해시(Hash)** | 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키로 변환하는 것을 말한다. |
| **양자 암호키 분배 (QKD; Quantum Key Distribution)** | 양자 통신을 위해 비밀키를 분배하여 관리하는 기술로, 두 시스템이 암호 알고리즘 동작을 위한 비밀키를 안전하게 공유하기 위해 양자 암호키 분배 시스템을 설치하여 운용하는 방식으로 활용된다. |
| **프라이버시 강화 기술 (PET; Privacy Enhancing Technology)** | • 개인정보 위험 관리 기술이다.<br>• PET는 최근 심각한 위험으로 대두되고 있는 개인정보 침해 위험을 관리하기 위한 핵심 기술로, 암호화, 익명화 등 개인정보를 보호하는 기술에서 사용자가 직접 개인정보를 통제하기 위한 기술까지 다양한 사용자 프라이버시 보호 기술을 통칭한다. |


| | |
|---|---|
| **디지털 저작권 관리(DRM; Digital Rights Management)** | 인터넷이나 기타 디지털 매체를 통해 유통되는 데이터의 저작권을 보호를 위해 데이터의 안전한 배포를 활성화하거나 불법 배포를 방지하기 위한 시스템이다. |
| **공통 평가 기준 (CC; Common Criteria)** | • 1999년 6월 8일 ISO 15408 표준으로 채택된 정보 보호 제품 평가 기준이다.<br>• 공통 평가 기준은 정보화 순기능 역할을 보장하기 위해 정보화 제품의 정보보호 기능과 이에 대한 사용 환경 등급을 정한 기준이다. |
| **개인정보 영향평가 제도 (PIA; Privacy Impact As sessment)** | 개인 정보를 활용하는 새로운 정보시스템의 도입 및 기존 정보시스템의 중요한 변경 시 시스템의 구축 · 운영이 기업의 고객은 물론 국민의 사생활에 미칠 영향에 대해 미리 조사 · 분석 · 평가하는 제도이다. |
| **그레이웨어 (Grayware)** | 소프트웨어를 제공하는 입장에서는 악의적이지 않은 유용한 소프트웨어라고 주장할 수 있지만 사용자 입장에서는 유용할 수도 있고 악의적일 수도 있는 애드웨어, 트랙웨어, 기타 악성 코드나 악성 공유웨어를 말한다. |
| **매시업(Mashup)** | 웹에서 제공하는 정보 및 서비스를 이용하여 새로운 소프트웨어나 서비스, 데이터베이스 등을 만드는 기술이다. 즉 다수의 정보원이 제공하는 콘텐츠를 조합하여 하나의 서비스로 제공하는 웹 사이트 또는 애플리케이션을 말한다. |
| **리치 인터넷 애플리케이션 (RIA; Rich Internet Application)** | 플래시 애니메이션 기술과 웹 서버 애플리케이션 기술을 통합하여 기존 HTML 보다 역동적이고 인터랙티브한 웹페이지를 제공하는 신개념의 플래시 웹페이지 제작 기술이다. |
| **시맨틱 웹 (Semantic Web)** | 컴퓨터가 사람을 대신하여 정보를 읽고 이해하고 가공하여 새로운 정보를 만들어 낼 수 있도록 이해하기 쉬운 의미를 가진 차세대 지능형 웹이다. |
| **증발품 (Vaporware)** | 판매 계획 또는 배포 계획은 발표되었으나 실제로 고객에게 판매되거나 배포되지 않고 있는 소프트웨어이다. |
| **오픈 그리드 서비스 아키텍처 (OGSA; Open Grid Service Architecture)** | 애플리케이션 공유를 위한 웹 서비스를 그리드 상에서 제공하기 위해 만든 개방형 표준이다. |
| **서비스 지향 아키텍처 (SOA; Service Oriented Architecture)** | • 기업의 소프트웨어 인프라인 정보시스템을 공유와 재사용이 가능한 서비스 단위나 컴포넌트 중심으로 구축하는 정보기술 아키텍처이다.<br>• SOA 기반 애플리케이션 구성 계층 : 표현(Presentation) 계층, 업무 프로세스(Biz-Process) 계층, 서비스 중간(Service Intermediary) 계층, 애플리케이션(Application) 계층, 데이터 저장(Persistency) 계층 |
| **서비스형 소프트웨어 (SaaS; Software as a Service)** | 소프트웨어의 여러 기능 중에서 사용자가 필요로 하는 서비스만 이용할 수 있도록 한 소프트웨어이다. |
| **소프트웨어 에스크로(임치) (Software Escrow)** | 소프트웨어 개발자의 지식재산권을 보호하고 사용자는 저렴한 비용으로 소프트웨어를 안정적으로 사용 및 유지보수 받을 수 있도록 소스 프로그램과 기술 정보 등을 제3의 기관에 보관하는 것이다. |
| **복잡 이벤트 처리 (CEP; Complex Event Processing)** | 실시간으로 발생하는 많은 사건들 중 의미가 있는 것만을 추출할 수 있도록 사건 발생 조건을 정의하는 데이터 처리 방법이다. |
| **디지털 트윈 (Digital Twin)** | • 현실속의 사물을 소프트웨어로 가상화한 모델로, 자동차, 항공, 에너지, 국방, 헬스케어 등 여러 분야에서 주목 받고 있다.<br>• 실제 물리적인 자산을 소프트웨어로 가상화함으로써 실제 자산의 특성에 대한 정확한 정보를 얻을 수 있고, 자산 최적화, 돌발사고 최소화, 생산성 증가 등 설계부터 제조, 서비스에 이르는 모든 과정의 효율성을 향상시킬 수 있다. |

## 핵심 337 소프트웨어 개발 직무별 보안 활동

안전한 소프트웨어 개발을 위해서는 프로젝트 관련자들이 보안 활동을 수행할 수 있도록 각 직무(Role)별로 수행해야 할 보안 활동을 정의해야 한다.

- 소프트웨어 개발 참여자의 직무는 프로젝트 관리자, 요구사항 분석가, 아키텍트, 설계자, 구현 개발자, 테스트 분석가, 보안 감사자로 구분한다.

| | |
|---|---|
| **프로젝트 관리자 (Project Manager)** | • 응용 프로그램에 대한 보안 전략을 조직 구성원들에게 전달한다.<br>• 조직 구성원들에게 응용 프로그램 보안 영향을 이해시킨다. |
| **요구사항 분석가 (Requirement Specifier)** | • 아키텍트가 고려해야 할 보안 관련 비즈니스 요구사항을 설명한다.<br>• 프로젝트 팀이 고려해야 할 구조 정의 및 해당 구조에 존재하는 자원에 대한 보안 요구사항을 정의한다. |

| | |
|---|---|
| 아키텍트 (Architect) | • 보안 오류가 발생하지 않도록 보안 기술 문제를 충분히 이해한다.<br>• 시스템에 사용되는 모든 리소스 정의 및 각 리소스별로 적절한 보안 요구사항을 적용한다. |
| 설계자 (Designer) | • 특정 기술에 대해 보안 요구사항의 만족성 여부를 확인한다.<br>• 문제 발생 시 최선의 문제 해결 방법을 결정한다.<br>• 애플리케이션 보안 수준에 대한 품질 측정을 지원한다. |
| 구현 개발자 (Implementer) | 구조화된 소프트웨어 개발 환경에서 프로그램을 원활히 구현할 수 있도록 시큐어 코딩 표준을 준수하여 개발한다. |
| 테스트 분석가 (Test Analyst) | • 소프트웨어 개발 요구사항과 구현 결과를 반복적으로 확인한다.<br>• 테스트 분석가는 반드시 보안 전문가일 필요는 없지만 보안 위험에 대한 학습이나 툴(Tool) 사용법 정도는 숙지하고 있어야 한다. |
| 보안 감사자 (Security Auditor) | • 소프트웨어 개발 프로젝트의 현재 상태의 보안을 보장한다.<br>• 요구사항 검토 시 요구사항의 적합성과 완전성을 확인한다.<br>• 소프트웨어 개발 프로젝트의 전체 단계에서 활동한다. |

## 핵심 338 개인정보 보호 관련 법령

- **개인정보 보호법** : 개인정보의 처리 및 보호에 관한 사항을 정함으로써 개인의 자유와 권리를 보호한다.
- **정보통신망 이용촉진 및 정보보호 등에 관한 법률** : 정보통신망의 이용 촉진 및 정보통신 서비스를 이용하는 이용자들의 개인정보를 보호한다.
- **신용정보의 이용 및 보호에 관한 법률** : 개인 신용정보의 효율적 이용과 체계적인 관리를 통해 정보의 오남용을 방지한다.
- **위치정보의 보호 및 이용 등에 관한 법률** : 개인 위치정보의 안전한 이용 환경을 조성하여 정보의 유출이나 오남용을 방지한다.
- **표준 개인정보 보호 지침** : 개인정보의 처리에 관한 기준, 개인정보 침해의 유형 및 예방조치 등에 관한 세부사항을 규정한다.
- **개인정보의 안전성 확보 조치 기준** : 개인정보 처리자가 개인정보를 처리하는데 있어 개인정보가 분실, 도난, 유출, 위조, 변조, 훼손되지 않도록 안전성 확보에 필요한 기술적, 관리적, 물리적 안전조치에 관한 최소한의 기준을 규정한다.
- **개인정보 영향평가에 관한 고시** : 개인정보 영향평가를 위한 평가기관의 지정, 영향평가의 절차 등에 관한 세부기준을 규정한다.

## 핵심 339 HW 관련 신기술

| | |
|---|---|
| 고가용성 (HA; High Availability) | • 긴 시간동안 안정적인 서비스 운영을 위해 장애 발생 시 즉시 다른 시스템으로 대체 가능한 환경을 구축하는 메커니즘을 의미한다.<br>• 가용성(Availability)을 극대화하는 방법으로는 클러스터, 이중화 등이 있다. |
| 3D Printing (Three Dimension Printing) | 대상을 평면에 출력하는 것이 아니라 손으로 만질 수 있는 실제 물체로 만들어내는 것을 말한다. |
| 4D Printing (Fourth Dimension Printing) | 특정 시간이나 환경 조건이 갖추어지면 스스로 형태를 변화시키거나 제조되는 자가 조립(Self-Assembly) 기술이 적용된 제품을 3D Printing하는 기술을 의미한다. |
| RAID(Redundant Array of Inexpensive Disk, Redundant Array of Independent Disk) | 여러 개의 하드디스크로 디스크 배열을 구성하여 파일을 구성하고 있는 데이터 블록들을 서로 다른 디스크들에 분산 저장할 경우 그 블록들을 여러 디스크에서 동시에 읽거나 쓸 수 있으므로 디스크의 속도가 매우 향상되는데, 이 기술을 RAID라고 한다. |
| 4K 해상도 | • 차세대 고화질 모니터의 해상도를 지칭하는 용어이다.<br>• 4K 해상도는 가로 픽셀 수가 3840이고, 세로 픽셀 수가 2160인 영상의 해상도를 말하는데, 이는 Full HDTV(1920×1080)의 가로·세로 2배, 총 4배에 해당하는 초고화질의 영상이다. |
| 앤 스크린 (N-Screen) | N개의 서로 다른 단말기에서 동일한 콘텐츠를 자유롭게 이용할 수 있는 서비스를 말한다. |
| 컴패니언 스크린 (Companion Screen) | 앤 스크린(N Screen)의 한 종류로, TV 방송 시청 시 방송 내용을 공유하며 추가적인 기능을 수행할 수 있는 스마트폰, 태블릿PC 등을 의미한다. 세컨드 스크린(Second Screen)이라고도 불린다. |


| | |
|---|---|
| 신 클라이언트 PC (Thin Client PC) | 하드디스크나 주변 장치 없이 기본적인 메모리만 갖추고 서버와 네트워크로 운용되는 개인용 컴퓨터를 말하는 것으로, 서버 기반 컴퓨팅과 관계가 깊다. |
| 패블릿(Phablet) | 폰(Phone)과 태블릿(Tablet)의 합성어로, 태블릿 기능을 포함한 5인치 이상의 대화면 스마트폰을 말한다. |
| C형 유에스비(Universal Serial Bus Type-C, USB Type-C, USB-C) | • 범용 인터페이스 규격인 유에스비(USB; Universal Serial Bus)의 표준 중 하나로, 2014년 8월 USB IF(Implementers Forum)에서 발표되었다.<br>• C형 유에스비는 기존 A형에 비하여 크기가 작고, 24핀으로 위아래의 구분이 없어 어느 방향으로든 연결이 가능하다. |
| 멤스(MEMS; Micro-Electro Mechanical Systems) | 초정밀 반도체 제조 기술을 바탕으로 센서, 액추에이터(Actuator) 등 기계 구조를 다양한 기술로 미세 가공하여 전기기계적 동작을 할 수 있도록 한 초미세 장치이다. |
| 트러스트존 기술 (TrustZone Technology) | 칩 설계회사인 ARM(Advanced RISC Machine)에서 개발한 기술로, 하나의 프로세서(Processor) 내에 일반 애플리케이션을 처리하는 일반 구역(Normal World)과 보안이 필요한 애플리케이션을 처리하는 보안 구역(Secure World)으로 분할하여 관리하는 하드웨어 기반의 보안 기술이다. |
| 엠디스크(M-DISC, Millennial DISC) | 한 번의 기록만으로 자료를 영구 보관할 수 있는 광 저장 장치이다. |
| 멤리스터(Memristor) | 메모리(Memory)와 레지스터(Resister)의 합성어로, 전류의 방향과 양 등 기존의 경험을 모두 기억하는 특별한 소자이다. |

## 핵심 340 Secure OS의 개요

Secure OS는 기존의 운영체제(OS)에 내재된 보안 취약점을 해소하기 위해 보안 기능을 갖춘 커널을 이식하여 외부의 침입으로부터 시스템 자원을 보호하는 운영체제를 의미한다.

- 보호 방법을 구현하기 복잡한 것부터 차례로 분류하면 다음과 같다.
  - 암호적 분리(Cryptographic Separation) : 내부 정보를 암호화하는 방법
  - 논리적 분리(Logical Separation) : 프로세스의 논리적 구역을 지정하여 구역을 벗어나는 행위를 제한하는 방법
  - 시간적 분리(Temporal Separation) : 동일 시간에 하나의 프로세스만 수행되도록 하여 동시 실행으로 발생하는 보안 취약점을 제거하는 방법
  - 물리적 분리(Physical Separation) : 사용자별로 특정 장비만 사용하도록 제한하는 방법

※ 참조 모니터(Reference Monitor)

참조 모니터는 보호대상의 객체에 대한 접근통제를 수행하는 추상머신이며, 이것을 실제로 구현한 것이 보안 커널이다.

- 참조 모니터는 보안 커널 데이터베이스(SKDB; Security Kernel Database)를 참조하여 객체에 대한 접근 허가 여부를 결정한다.
- 참조 모니터와 보안 커널의 특징
  - 격리성(Isolation) : 부정 조작이 불가능해야 함
  - 검증가능성(Verifiability) : 적절히 구현되었다는 것을 확인할 수 있어야 함
  - 완전성(Completeness) : 우회가 불가능해야 함

## 핵심 341 DB 관련 신기술

| | |
|---|---|
| 빅데이터 (Big Data) | 기존의 관리 방법이나 분석 체계로는 처리하기 어려운 막대한 양의 정형 또는 비정형 데이터 집합으로, 스마트 단말의 빠른 확산, 소셜 네트워크 서비스의 활성화, 사물 네트워크의 확대로 데이터 폭발이 더욱 가속화되고 있다. |
| 브로드 데이터 (Broad Data) | 다양한 채널에서 소비자와 상호 작용을 통해 생성된, 기업 마케팅에 있어 효율적이고 다양한 데이터이며, 이전에 사용하지 않거나 알지 못했던 새로운 데이터나, 기존 데이터에 새로운 가치가 더해진 데이터를 말한다. |
| 메타 데이터 (Meta Data) | 일련의 데이터를 정의하고 설명해 주는 데이터이다. 컴퓨터에서는 데이터 사전의 내용, 스키마 등을 의미하고, HTML 문서에서는 메타 태그 내의 내용이 메타 데이터이다. 방송에서는 방대한 분량의 저작물을 신속하게 검색하기 위한 촬영 일시, 장소, 작가, 출연자 등과 음원의 검색을 위한 작곡자나 가수명 등을 메타 데이터로 처리한다. |
| 디지털 아카이빙 (Digital Archiving) | 디지털 정보 자원을 장기적으로 보존하기 위한 작업을 말한다. 아날로그 콘텐츠는 디지털로 변환한 후 압축해서 저장하고, 디지털 콘텐츠도 체계적으로 분류하고 메타 데이터를 만들어 DB화하는 작업이다. |

| | |
|---|---|
| 하둡 (Hadoop) | • 오픈 소스를 기반으로 한 분산 컴퓨팅 플랫폼이다.<br>• 일반 PC급 컴퓨터들로 가상화된 대형 스토리지를 형성하고 그 안에 보관된 거대한 데이터 세트를 병렬로 처리할 수 있도록 개발된 자바 소프트웨어 프레임워크로, 구글, 야후 등에 적용되고 있다. |
| 맵리듀스 (MapReduce) | 대용량 데이터를 분산 처리하기 위한 목적으로 개발된 프로그래밍 모델로, 흩어져 있는 데이터를 연관성 있는 데이터 분류로 묶는 Map 작업을 수행한 후 중복 데이터를 제거하고 원하는 데이터를 추출하는 Reduce 직업을 수행한다. |
| 타조(Tajo) | 오픈 소스 기반 분산 컴퓨팅 플랫폼인 아파치 하둡(Apache Hadoop) 기반의 분산 데이터 웨어하우스 프로젝트로, 우리나라가 주도하여 개발하고 있다. |
| 데이터 다이어트 (Data Diet) | 데이터를 삭제하는 것이 아니라 압축하고, 중복된 정보는 중복을 배제하고, 새로운 기준에 따라 나누어 저장하는 작업이다. |
| 데이터 마이닝 (Data Mining) | • 데이터 웨어하우스에 저장된 데이터 집합에서 사용자의 요구에 따라 유용하고 가능성 있는 정보를 발견하기 위한 기법이다.<br>• 대량의 데이터를 분석하여 데이터 속에 내재되어 있는 변수 사이의 상호관계를 규명하여 패턴화함으로써 효율적인 데이터 추출이 가능하다. |
| OLAP(Online Analytical Processing) | • 다차원으로 이루어진 데이터로부터 통계적인 요약 정보를 분석하여 의사결정에 활용하는 방식을 말한다.<br>• OLAP 연산 : Roll-up, Drill-down, Drill-through, Drill-across, Pivoting, Slicing, Dicing |

## 핵심 342 회복(Recovery)

• 회복은 트랜잭션들을 수행하는 도중 장애가 발생하여 데이터베이스가 손상되었을 때 손상되기 이전의 정상 상태로 복구하는 작업이다.

• 회복 기법

| | |
|---|---|
| 연기 갱신 기법 (Deferred Update) | • 트랜잭션이 성공적으로 완료될 때까지 데이터베이스에 대한 실질적인 갱신을 연기하는 방법이다.<br>• 트랜잭션이 수행되는 동안 갱신된 내용은 일단 Log에 보관된다.<br>• 트랜잭션의 부분 완료(성공적인 완료 직전) 시점에 Log에 보관한 갱신 내용을 실제 데이터베이스에 기록한다.<br>• 트랜잭션이 부분 완료되기 전에 장애가 발생하여 트랜잭션이 Rollback되면 트랜잭션이 실제 데이터베이스에 영향을 미치지 않았기 때문에 어떠한 갱신 내용도 취소(Undo)시킬 필요 없이 무시하면 된다.<br>• Redo 작업만 가능하다. |
| 즉각 갱신 기법 (Immediate Update) | • 트랜잭션이 데이터를 갱신하면 트랜잭션이 부분 완료되기 전이라도 즉시 실제 데이터베이스에 반영하는 방법이다.<br>• 장애가 발생하여 회복 작업할 경우를 대비하여 갱신된 내용들은 Log에 보관시킨다.<br>• 회복 작업을 할 경우에는 Redo와 Undo 모두 사용 가능하다. |
| 그림자 페이지 대체 기법 (Shadow Paging) | • 갱신 이전의 데이터베이스를 일정 크기의 페이지 단위로 구성하여 각 페이지마다 복사본인 그림자 페이지로 별도 보관해 놓고, 실제 페이지를 대상으로 트랜잭션에 의한 갱신 작업을 하다가 장애가 발생하여 트랜잭션 작업을 Rollback시킬 때, 갱신된 이후의 실제 페이지 부분에 그림자 페이지를 대체하여 회복시키는 기법이다.<br>• 로그, Undo 및 Redo 알고리즘이 필요 없다. |
| 검사점 기법 (Check Point) | 트랜잭션 실행 중 특정 단계에서 재실행할 수 있도록 갱신 내용이나 시스템에 대한 상황 등에 관한 정보와 함께 검사점을 로그에 보관해 두고, 장애 발생 시 트랜잭션 전체를 철회하지 않고 검사점부터 회복 작업을 하여 회복시간을 절약하도록 하는 기법이다. |

## 핵심 343 병행제어(Concurrency Control)

병행제어란 다중 프로그램의 이점을 활용하여 동시에 여러 개의 트랜잭션을 병행수행할 때, 동시에 실행되는 트랜잭션들이 데이터베이스의 일관성을 파괴하지 않도록 트랜잭션 간의 상호 작용을 제어하는 것이다.

• 병행제어 기법의 종류

| | |
|---|---|
| 로킹(Locking) | • 주요 데이터의 액세스를 상호 배타적으로 하는 것이다.<br>• 트랜잭션들이 어떤 로킹 단위를 액세스하기 전에 Lock(잠금)을 요청해서 Lock이 허락되어야만 그 로킹 단위를 액세스할 수 있도록 하는 기법이다. |
| 타임 스탬프 순서 (Time Stamp Ordering) | • 직렬성 순서를 결정하기 위해 트랜잭션 간의 처리 순서를 미리 선택하는 기법들 중에서 가장 보편적인 방법이다.<br>• 트랜잭션과 트랜잭션이 읽거나 갱신한 데이터에 대해 트랜잭션이 실행을 시작하기 전에 시간표(Time Stamp)를 부여하여 부여된 시간에 따라 트랜잭션 작업을 수행하는 기법이다.<br>• 교착상태가 발생하지 않는다. |

| | |
|---|---|
| 최적 병행수행 (검증 기법, 확인 기법, 낙관적 기법) | 병행수행하고자 하는 대부분의 트랜잭션이 판독 전용(Read Only) 트랜잭션일 경우, 트랜잭션 간의 충돌률이 매우 낮아서 병행제어 기법을 사용하지 않고 실행되어도 이 중의 많은 트랜잭션은 시스템의 상태를 일관성 있게 유지한다는 점을 이용한 기법이다. |
| 다중 버전 기법 | • 타임 스탬프의 개념을 이용하는 기법으로, 다중 버전 타임 스탬프 기법이라고도 한다.<br>• 타임 스탬프 기법은 트랜잭션 및 데이터들이 이용될 때의 시간을 시간표로 관리하지만, 다중 버전 기법은 갱신될 때마다의 버전을 부여하여 관리한다. |

※ 로킹 단위(Locking Granularity)

- 병행제어에서 한꺼번에 로킹할 수 있는 객체의 크기를 의미한다.
- 데이터베이스, 파일, 레코드, 필드 등이 로킹 단위가 될 수 있다.
- 로킹 단위가 크면 로크 수가 작아 관리하기 쉽지만 병행성 수준이 낮아지고, 로킹 단위가 작으면 로크 수가 많아 관리하기 복잡해 오버헤드가 증가하지만 병행성 수준이 높아진다.

## 핵심 344 교착상태

교착상태(Dead Lock)는 상호 배제에 의해 나타나는 문제점으로, 둘 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 점유하고 있는 자원을 요구하며 무한정 기다리는 현상을 의미한다.

- 교착상태 발생의 필요 충분 조건

| | |
|---|---|
| 상호 배제 (Mutual Exclusion) | 한 번에 한 개의 프로세스만이 공유 자원을 사용할 수 있어야 한다. |
| 점유와 대기 (Hold and Wait) | 최소한 하나의 자원을 점유하고 있으면서 다른 프로세스에 할당되어 사용되고 있는 자원을 추가로 점유하기 위해 대기하는 프로세스가 있어야 한다. |
| 비선점 (Non-preemption) | 다른 프로세스에 할당된 자원은 사용이 끝날 때까지 강제로 빼앗을 수 없어야 한다. |
| 환형 대기 (Circular Wait) | 공유 자원과 공유 자원을 사용하기 위해 대기하는 프로세스들이 원형으로 구성되어 있어 자신에게 할당된 자원을 점유하면서 앞이나 뒤에 있는 프로세스의 자원을 요구해야 한다. |

- 교착상태의 해결 방법

| | |
|---|---|
| 예방 기법 (Prevention) | • 교착상태가 발생하지 않도록 사전에 시스템을 제어하는 방법으로, 교착상태 발생의 네 가지 조건 중에서 어느 하나를 제거(부정)함으로써 수행된다.<br>• 자원의 낭비가 가장 심한 기법이다. |
| 회피 기법 (Avoidance) | • 교착상태가 발생할 가능성을 배제하지 않고 교착상태가 발생하면 적절히 피해나가는 방법으로, 주로 은행원 알고리즘(Banker's Algorithm)이 사용된다.<br>• 은행원 알고리즘(Banker's Algorithm) : E. J. Dijkstra가 제안한 것으로, 은행에서 모든 고객의 요구가 충족되도록 현금을 할당하는 데서 유래한 기법 |
| 발견 기법 (Detection) | • 시스템에 교착상태가 발생했는지 점검하여 교착상태에 있는 프로세스와 자원을 발견하는 것을 의미한다.<br>• 교착상태 발견 알고리즘과 자원 할당 그래프 등을 사용할 수 있다. |
| 회복 기법 (Recovery) | 교착상태를 일으킨 프로세스를 종료하거나 교착상태의 프로세스에 할당된 자원을 선점하여 프로세스나 자원을 회복하는 것을 의미한다. |

## 핵심 345 데이터 표준화

데이터 표준화는 시스템을 구성하는 데이터 요소의 명칭, 정의, 형식, 규칙에 대한 원칙을 수립하고 적용하는 것을 의미한다.

- 데이터 표준화의 대상으로는 데이터 명칭, 데이터 정의, 데이터 형식, 데이터 규칙이 있다.
- 데이터 표준화의 구성 요소에는 데이터 표준, 데이터 관리 조직, 데이터 표준화 절차가 있다.

| | |
|---|---|
| 데이터 표준 | • 데이터 모델이나 DB에서 정의할 수 있는 모든 오브젝트를 대상으로 데이터 표준화를 수행해야 한다.<br>• 데이터 표준의 종류<br>– 표준 단어 : 업무에서 사용하고 일정한 의미를 갖고 있는 최소 단위의 단어를 의미한다.<br>– 표준 도메인 : 문자형, 숫자형, 날짜형, 시간형과 같이 컬럼을 성질에 따라 그룹핑한 개념이다.<br>– 표준 코드 : 선택할 수 있는 값을 정형화하기 위해 기준에 맞게 이미 정의된 코드 값으로, 도메인의 한 유형이다.<br>– 표준 용어 : 단어, 도메인, 코드 표준이 정의되면 이를 바탕으로 표준 용어를 구성한다. |


| | |
|---|---|
| 데이터 관리 조직 | • 데이터 표준 원칙이나 데이터 표준의 준수 여부 등을 관리하는 사람들로, 대표적으로 데이터 관리자가 있다.<br>• 데이터 관리자는 조직 내의 데이터에 대한 정의, 체계화, 감독 등의 업무를 담당한다. |
| 데이터 표준화 절차 | 데이터 표준화 요구사항 수집, 데이터 표준 정의, 데이터 표준 확정, 데이터 표준 관리 순으로 진행된다. |

- 데이터 표준화의 기대 효과
  - 동일한 데이터에 대해 동일한 명칭을 지정하면 명확한 의사소통이 가능하다.
  - 표준화된 데이터를 사용하면 필요한 데이터의 의미나 위치 등을 쉽게 파악할 수 있다.
  - 데이터 표준에 따라 데이터 형식 및 규칙을 적용하면 입력 오류를 방지하고 잘못된 데이터로 인한 의사 결정의 오류를 줄여 데이터 품질을 향상시킬 수 있다.
  - 데이터 표준에 따라 데이터를 전사적으로 관리하면 시스템 간 데이터 공유 시 데이터 변환이나 정제 작업을 수행하지 않아도 된다.
  - 향후 데이터 유지보수 및 운영의 효율성, 관리 비용을 절감할 수 있다.

## 핵심 346 Secure SDLC의 개요

Secure SDLC는 보안상 안전한 소프트웨어를 개발하기 위해 SDLC에 보안 강화를 위한 프로세스를 포함한 것을 의미한다.

- Secure SDLC는 소프트웨어의 유지 보수 단계에서 보안 이슈를 해결하기 위해 소모되는 많은 비용을 최소화하기 위해 등장하였다.
- Secure SDLC는 요구사항 분석, 설계, 구현, 테스트, 유지 보수 등 SDLC 전체 단계에 걸쳐 수행되어야 할 보안 활동을 제시한다.
- Secure SDLC의 대표적인 방법론

| | |
|---|---|
| CLASP | • Secure Software 사에서 개발하였으며, SDLC의 초기 단계에서 보안을 강화하기 위해 개발된 방법론이다.<br>• 활동 중심, 역할 기반의 프로세스로 구성되어 있으며, 현재 운용 중인 시스템에 적용하기에 적합하다. |
| SDL | • 마이크로소프트 사에서 안전한 소프트웨어 개발을 위해 기존의 SDLC를 개선한 방법론이다.<br>• 전통적인 나선형 모델을 기반으로 한다. |
| Seven Touchpoints | • 소프트웨어 보안의 모범사례를 SDLC에 통합한 방법론이다.<br>• 설계 및 개발 과정의 모든 산출물에 대해 위험 분석 및 테스트를 수행한다.<br>• SDLC의 각 단계에 관련된 7개의 보안 강화 활동을 수행한다. |

## 핵심 347 보안 요소

보안 요소는 소프트웨어 개발에 있어 충족시켜야할 요소 및 요건을 의미한다.

- 보안 3대 요소에는 기밀성(Confidentiality), 무결성(Integrity), 가용성(Availability)이 있으며, 그 외에도 인증(Authentication), 부인 방지(NonRepudiation) 등이 있다.

| | |
|---|---|
| 기밀성 | • 시스템 내의 정보와 자원은 인가된 사용자에게만 접근이 허용된다.<br>• 정보가 전송 중에 노출되더라도 데이터를 읽을 수 없다. |
| 무결성 | 시스템 내의 정보는 오직 인가된 사용자만 수정할 수 있다. |
| 가용성 | 인가받은 사용자는 언제라도 사용할 수 있다. |
| 인증 | • 시스템 내의 정보와 자원을 사용하려는 사용자가 합법적인 사용자인지를 확인하는 모든 행위를 말한다.<br>• 대표적 방법으로는 패스워드, 인증용 카드, 지문 검사 등이 있다. |
| 부인 방지 | 데이터를 송 · 수신한 자가 송 · 수신 사실을 부인할 수 없도록 송 · 수신 증거를 제공한다. |

## 핵심 348 세션 통제

### 세션 통제의 개요

세션은 서버와 클라이언트의 연결을 의미하고, 세션 통제는 세션의 연결과 연결로 인해 발생하는 정보를 관리하는 것을 의미한다.

- 세션 통제는 소프트웨어 개발 과정 중 요구사항 분석 및 설계 단계에서 진단해야 하는 보안 점검 내용이다.
- 세션 통제의 보안 약점에는 불충분한 세션 관리, 잘못된 세션에 의한 정보 노출이 있다.


### 세션 설계시 고려 사항

- 시스템의 모든 페이지에서 로그아웃이 가능하도록 UI(User Interface)를 구성한다.
- 로그아웃 요청 시 할당된 세션이 완전히 제거되도록 한다.
- 세션 타임아웃은 중요도가 높으면 2~5분, 낮으면 15~30분으로 설정한다.
- 이전 세션이 종료되지 않으면 새 세션이 생성되지 못하도록 설계한다.
- 중복 로그인을 허용하지 않은 경우 클라이언트의 중복 접근에 대한 세션 관리 정책을 수립한다.
- 패스워드 변경 시 활성화된 세션을 삭제하고 재할당한다.

### 세션ID의 관리 방법

- 세션ID는 안전한 서버에서 최소 128비트의 길이로 생성한다.
- 세션ID의 예측이 불가능하도록 안전한 난수 알고리즘을 적용한다.
- 세션ID가 노출되지 않도록 URL Rewrite 기능을 사용하지 않는 방향으로 설계한다.
- 로그인 시 로그인 전의 세션ID를 삭제하고 재할당한다.
- 장기간 접속하고 있는 세션ID는 주기적으로 재할당되도록 설계한다.

## 핵심 349 입력 데이터 검증 및 표현

### 입력 데이터 검증 및 표현의 개요

입력 데이터 검증 및 표현은 입력 데이터로 인해 발생하는 문제들을 예방하기 위해 구현 단계에서 검증해야 하는 보안 점검 항목들이다.

### 입력 데이터 검증 및 표현의 보안 약점

| | |
|---|---|
| SQL 삽입 | • 웹 응용 프로그램에 SQL을 삽입하여 내부 데이터베이스(DB) 서버의 데이터를 유출 및 변조하고, 관리자 인증을 우회하는 보안 약점이다.<br>• 동적 쿼리에 사용되는 입력 데이터에 예약어 및 특수문자가 입력되지 않게 필터링 되도록 설정하여 방지할 수 있다. |
| 경로 조작 및 자원 삽입 | • 데이터 입출력 경로를 조작하여 서버 자원을 수정 · 삭제할 수 있는 보안 약점이다.<br>• 사용자 입력값을 식별자로 사용하는 경우, 경로 순회 공격을 막는 필터를 사용하여 방지할 수 있다. |
| 크로스사이트 스크립팅(XSS) | • 웹페이지에 악의적인 스크립트를 삽입하여 방문자들의 정보를 탈취하거나, 비정상적인 기능 수행을 유발하는 보안 약점이다.<br>• HTML 태그의 사용을 제한하거나 스크립트에 삽입되지 않도록 '〈', '〉', '&' 등의 문자를 다른 문자로 치환함으로써 방지할 수 있다. |
| 운영체제 명령어 삽입 | • 외부 입력값을 통해 시스템 명령어의 실행을 유도함으로써 권한을 탈취하거나 시스템 장애를 유발하는 보안 약점이다.<br>• 웹 인터페이스를 통해 시스템 명령어가 전달되지 않도록 하고, 외부 입력값을 검증 없이 내부 명령어로 사용하지 않음으로써 방지할 수 있다. |
| 위험한 형식 파일 업로드 | • 악의적인 명령어가 포함된 스크립트 파일을 업로드함으로써 시스템에 손상을 주거나, 시스템을 제어할 수 있는 보안 약점이다.<br>• 업로드 되는 파일의 확장자 제한, 파일명의 암호화, 웹사이트와 파일 서버의 경로 분리, 실행 속성을 제거하는 등의 방법으로 방지할 수 있다. |
| 신뢰되지 않는 URL 주소로 자동접속 연결 | • 입력 값으로 사이트 주소를 받는 경우 이를 조작하여 방문자를 피싱 사이트로 유도하는 보안 약점이다.<br>• 연결되는 외부 사이트의 주소를 화이트 리스트로 관리함으로써 방지할 수 있다. |
| 메모리 버퍼 오버플로 | • 연속된 메모리 공간을 사용하는 프로그램에서 할당된 메모리의 범위를 넘어선 위치에서 자료를 읽거나 쓰려고 할 때 발생하는 보안 약점이다.<br>• 프로그램의 오동작을 유발시키거나, 악의적인 코드를 실행시켜 공격자가 프로그램을 통제할 수 있는 권한을 획득하게 한다.<br>• 메모리 버퍼를 사용할 경우 적절한 버퍼의 크기를 설정하고, 설정된 범위의 메모리 내에서 올바르게 읽거나 쓸 수 있도록 함으로써 방지할 수 있다. |

## 핵심 350 보안 기능

### 보안 기능의 개요

보안 기능은 소프트웨어 개발의 구현 단계에서 코딩하는 기능인 인증, 접근제어, 기밀성, 암호화 등을 올바르게 구현하기 위한 보안 점검 항목들이다.

### 보안 기능의 보안 약점

| | |
|---|---|
| 적절한 인증 없이 중요기능 허용 | • 보안검사를 우회하여 인증과정 없이 중요한 정보 또는 기능에 접근 및 변경이 가능하다.<br>• 중요정보나 기능을 수행하는 페이지에서는 재인증 기능을 수행하도록 하여 방지할 수 있다. |


| | |
|---|---|
| 부적절한 인가 | • 접근제어 기능이 없는 실행경로를 통해 정보 또는 권한을 탈취할 수 있다.<br>• 모든 실행경로에 대해 접근제어 검사를 수행하고, 사용자에게는 반드시 필요한 접근 권한만을 부여하여 방지할 수 있다. |
| 중요한 자원에 대한 잘못된 권한 설정 | • 권한 설정이 잘못된 자원에 접근하여 해당 자원을 임의로 사용할 수 있다.<br>• 소프트웨어 관리자만 자원들을 읽고 쓸 수 있도록 설정하고, 인가되지 않은 사용자의 중요 자원에 대한 접근 여부를 검사함으로써 방지할 수 있다. |
| 취약한 암호화 알고리즘 사용 | • 암호화된 환경설정 파일을 해독하여 비밀번호 등의 중요정보를 탈취할 수 있다.<br>• 안전한 암호화 알고리즘을 이용하고, 업무관련 내용이나 개인정보 등에 대해서는 IT보안인증사무국이 안정성을 확인한 암호모듈을 이용함으로써 방지할 수 있다. |
| 중요정보 평문 저장 및 전송 | • 암호화되지 않은 평문 데이터를 탈취하여 중요한 정보를 획득할 수 있다.<br>• 중요한 정보를 저장하거나 전송할 때는 반드시 암호화 과정을 거치도록 하고, HTTPS 또는 SSL과 같은 보안 채널을 이용함으로써 방지할 수 있다. |
| 하드코드된 비밀번호 | • 소스코드 유출 시 내부에 하드코드된 패스워드를 이용하여 관리자 권한을 탈취할 수 있다.<br>• 패스워드는 암호화하여 별도의 파일에 저장하고, 디폴트 패스워드나 디폴드 키의 사용을 피함으로써 방지할 수 있다. |

## 핵심 351 시간 및 상태

### 시간 및 상태의 개요

시간 및 상태는 동시 수행을 지원하는 병렬 처리 시스템이나 다수의 프로세스가 동작하는 환경에서 시간과 실행 상태를 관리하여 시스템이 원활하게 동작되도록 하기 위한 보안 검증 항목들이다.

| | |
|---|---|
| TOCTOU 경쟁 조건 | • 검사 시점(Time Of Check)과 사용 시점(Time Of Use)을 고려하지 않고 코딩하는 경우 발생하는 보안 약점이다.<br>• 검사 시점에는 사용이 가능했던 자원이 사용 시점에는 사용할 수 없게 된 경우에 발생한다. |
| 종료되지 않는 반복문 또는 재귀함수 | • 반복문이나 재귀함수에서 종료 조건을 정의하지 않았거나 논리 구조상 종료될 수 없는 경우 발생하는 보안 약점이다.<br>• 반복문이나 재귀함수가 종료되지 않을 경우 시스템 자원이 끊임없이 사용되어 자원고갈로 인한 서비스 장애 또는 시스템 장애가 발생한다. |

## 핵심 352 에러처리

### 에러처리의 개요

에러처리는 소프트웨어 실행 중 발생할 수 있는 오류(Error)들을 사전에 정의하여 오류로 인해 발생할 수 있는 문제들을 예방하기 위한 보안 점검 항목들이다.

- 각 프로그래밍 언어의 예외처리 구문을 통해 오류에 대한 사항을 정의한다.
- 예외처리 구문으로 처리하지 못한 오류들은 중요정보를 노출시키거나, 소프트웨어의 실행이 중단되는 등 예기치 못한 문제를 발생시킬 수 있다.
- 에러처리의 미비로 인한 코딩이 유발하는 보안 약점에는 오류 메시지를 통한 정보노출, 오류 상황 대응 부재, 부적절한 예외처리가 있다.

| | |
|---|---|
| 오류 메시지를 통한 정보노출 | 오류 발생으로 실행 환경, 사용자 정보, 디버깅 정보 등의 중요 정보를 소프트웨어가 메시지로 외부에 노출하는 보안 약점이다. |
| 오류 상황 대응 부재 | 소프트웨어 개발 중 예외처리를 하지 않았거나 미비로 인해 발생하는 보안 약점이다. |
| 부적절한 예외처리 | 함수의 반환값 또는 오류들을 세분화하여 처리하지 않고 광범위하게 묶어 한 번에 처리하거나, 누락된 예외가 존재할 때 발생하는 보안 약점이다. |

## 핵심 353 코드 오류

### 코드 오류의 개요

코드 오류는 소프트웨어 구현 단계에서 개발자들이 코딩 중 실수하기 쉬운 형(Type) 변환, 자원 반환 등의 오류를 예방하기 위한 보안 점검 항목들이다.

| | |
|---|---|
| 널 포인터(Null Pointer) 역참조 | 널 포인터가 가리키는 메모리에 어떠한 값을 저장할 때 발생하는 보안 약점이다. |
| 부적절한 자원 해제 | 자원을 반환하는 코드를 누락하거나 프로그램 오류로 할당된 자원을 반환하지 못했을 때 발생하는 보안 약점이다. |
| 해제된 자원 사용 | 이미 사용이 종료되어 반환된 메모리를 참조하는 경우 발생하는 보안 약점이다. |
| 초기화되지 않은 변수 사용 | 변수 선언 후 값이 부여되지 않은 변수를 사용할 때 발생하는 보안 약점이다. |

※ 스택 가드(Stack Guard)

- 널 포인터 역참조와 같이 주소가 저장되는 스택에서 발생하는 보안 약점을 막는 기술 중 하나이다.
- 메모리상에서 프로그램의 복귀 주소와 변수 사이에 특정 값을 저장한 후 그 값이 변경되었을 경우 오버플로우 상태로 판단하여 프로그램 실행을 중단함으로써 잘못된 복귀 주소의 호출을 막는 기술이다.

## 핵심 354 캡슐화

### 캡슐화의 개요

캡슐화는 정보 은닉이 필요한 중요한 데이터와 기능을 불충분하게 캡슐화하거나 잘못 사용함으로써 발생할 수 있는 문제를 예방하기 위한 보안 점검 항목들이다.

| | |
|---|---|
| **잘못된 세션에 의한 정보 노출** | 다중 스레드(Multi-Thread) 환경에서 멤버 변수에 정보를 저장할 때 발생하는 보안 약점이다. |
| **제거되지 않고 남은 디버그 코드** | 개발 중에 버그 수정이나 결과값 확인을 위해 남겨둔 코드들로 인해 발생하는 보안 약점이다. |
| **시스템 데이터 정보 노출** | 시스템의 내부 정보를 시스템 메시지 등을 통해 외부로 출력하도록 코딩했을 때 발생하는 보안 약점이다. |
| **Public 메소드로부터 반환된 Private 배열** | 선언된 클래스 내에서만 접근이 가능한 Private 배열을 모든 클래스에서 접근이 가능한 Public 메소드에서 반환할 때 발생하는 보안 약점이다. |
| **Private 배열에 Public 데이터 할당** | Private 배열에 Public으로 선언된 데이터 또는 메소드의 파라미터를 저장할 때 발생하는 보안 약점이다. |

※ 접근 지정자(접근 제어자)

접근 지정자는 프로그래밍 언어에서 특정 개체를 선언할 때 외부로부터의 접근을 제한하기 위해 사용되는 예약어이다(접근 가능 : ○, 접근 불가능 : ×).

| 한정자 | 클래스 내부 | 패키지 내부 | 하위 클래스 | 패키지 외부 |
|---|---|---|---|---|
| Public | ○ | ○ | ○ | ○ |
| Protected | ○ | ○ | ○ | × |
| Default | ○ | ○ | × | × |
| Private | ○ | × | × | × |

## 핵심 355 API 오용

### API 오용의 개요

API 오용은 소프트웨어 구현 단계에서 API를 잘못 사용하거나 보안에 취약한 API를 사용하지 않도록 하기 위한 보안 검증 항목들이다.

### DNS Lookup에 의존한 보안 결정

도메인명에 의존하여 인증이나 접근 통제 등의 보안 결정을 내리는 경우 발생하는 보안 약점이다.

- DNS 엔트리를 속여 동일한 도메인에 속한 서버인 것처럼 위장하거나, 사용자와 서버 간의 네트워크 트래픽을 유도하여 악성 사이트를 경유하도록 조작할 수 있다.

### 취약한 API 사용

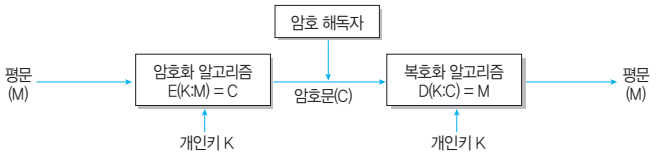
보안 문제로 사용이 금지된 API를 사용하거나, 잘못된 방식으로 API를 사용했을 때 발생하는 보안 약점이다.

- 보안 문제로 금지된 대표적인 API에는 C언어의 문자열 함수 strcat( ), strcpy( ), sprintf( ) 등이 있다.

## 핵심 356 개인키 암호화(Private Key Encryption) 기법

개인키 암호화 기법은 동일한 키로 데이터를 암호화하고 복호화한다.

- 데이터베이스 사용자는 평문의 정보 M을 암호화 알고리즘 E와 개인키(Private Key) K를 이용하여 암호문 C로 바꾸어 저장시켜 놓으면 사용자는 그 데이터베이스에 접근하기 위해 복호화 알고리즘 D와 개인키 K를 이용하여 다시 평문의 정보 M으로 바꾸어 이용하는 방법이다.



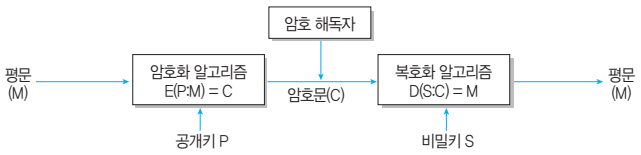

- 개인키 암호화 기법은 대칭 암호 기법 또는 단일키 암호화 기법이라고도 한다.
- 개인키 암호화 기법은 한 번에 하나의 데이터 블록을 암호화 하는 블록 암호화 방식과, 평문과 동일한 길이의 스트림을 생성하여 비트 단위로 암호화 하는 스트림 암호화 방식으로 분류된다.

- 종류
  - 블록 암호화 방식 : DES, SEED, AES, ARIA
  - 스트림 암호화 방식 : LFSR, RC4

## 핵심 357 공개키 암호화(Public Key Encryption) 기법

공개키 암호화 기법은 데이터를 암호화할 때 사용하는 공개키(Public Key)는 데이터베이스 사용자에게 공개하고, 복호화할 때의 비밀키(Secret Key)는 관리자가 비밀리에 관리한다.

- 데이터베이스 사용자는 평문의 정보 M을 암호화 알고리즘 E와 공개키(Public Key) P를 이용하여 암호문 C로 바꾸어 저장시켜 놓고, 이를 복호화하기 위해서는 비밀키와 복호화 알고리즘에 권한이 있는 사용자만이 복호화 알고리즘 D와 비밀키(Secret Key) S를 이용하여 다시 평문의 정보 M으로 바꿀 수 있는 기법이다.




- 공개키 암호화 기법은 비대칭 암호 기법이라고도 하며, 대표적으로는 RSA(Rivest Shamir Adleman) 기법이 있다.

## 핵심 358 양방향 알고리즘 종류

개인키 암호화 방식과 공개키 암호화 방식에서 사용되는 주요 암호화 알고리즘에는 SEED, ARIA 등이 있다.

| | |
|---|---|
| SEED | • 1999년 한국인터넷진흥원(KISA)에서 개발한 블록 암호화 알고리즘<br>• 블록 크기는 128비트이며, 키 길이에 따라 128, 256으로 분류된다. |
| ARIA(Academy, Research Institute, Agency) | • 2004년 국가정보원과 산학연협회가 개발한 블록 암호화 알고리즘<br>• ARIA는 학계(Academy), 연구기관(Research Institute), 정부(Agency)의 영문 앞 글자로 구성되었다.<br>• 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류된다. |
| DES(Data Encryption Standard) | • 1975년 미국 NBS에서 발표한 개인키 암호화 알고리즘<br>• DES를 3번 적용하여 보안을 더욱 강화한 3DES(Triple DES)도 있다.<br>• 블록 크기는 64비트이며, 키 길이는 56비트이다. |
| AES(Advanced Encryption Standard) | • 2001년 미국 표준 기술 연구소(NIST)에서 발표한 개인키 암호화 알고리즘<br>• DES의 한계를 느낀 NIST에서 공모한 후 발표하였다.<br>• 블록 크기는 128비트이며, 키 길이에 따라 128, 192, 256으로 분류된다. |
| RSA(Rivest Shamir Adleman) | • 1978년 MIT의 라이베스트(Rivest), 샤미르(Shamir), 애들먼(Adelman)에 의해 제안된 공개키 암호화 알고리즘<br>• 큰 숫자를 소인수분해 하기 어렵다는 것에 기반하여 만들어졌다.<br>• 공개키와 비밀키를 사용하는데, 여기서 키란 메시지를 열고 잠그는 상수(Constant)를 의미한다. |

## 핵심 359 해시(Hash)

해시는 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키로 변환하는 것을 의미한다.

- 해시 알고리즘을 해시 함수라고 부르며, 해시 함수로 변환된 값이나 키를 해시값 또는 해시키라고 부른다.
- 데이터의 암호화, 무결성 검증을 위해 사용될 뿐만 아니라 정보보호의 다양한 분야에서 활용된다.
- 해시 함수의 종류에는 SHA 시리즈, MD5, N-NASH, SNEFRU 등이 있다.

| | |
|---|---|
| SHA 시리즈 | • 1993년 미국 국가안보국(NSA)이 처음 설계했으며, 미국 국립표준기술연구소(NIST)에 의해 발표되었다.<br>• 초기 개발된 SHA-0 이후 SHA-1이 발표되었고, 다시 SHA-2라고 불리는 SHA-224, SHA-256, SHA-384, SHA-512가 발표되었다. |
| MD5 | • 1991년 R.Rivest가 MD4를 대체하기 위해 고안한 암호화 해시 함수이다.<br>• 블록 크기는 512비트이며, 키 길이는 128비트이다. |
| N-NASH | • 1989년 일본의 전신전화주식회사(NTT)에서 발표한 암호화 해시 함수이다.<br>• 블록 크기와 키 길이가 모두 128비트이다. |
| SNEFRU | • 1990년 R.C.Merkle가 발표한 해시 함수이다.<br>• 32비트 프로세서에서 구현을 용이하게 할 목적으로 개발되었다.<br>• 블록 크기는 512비트이며, 키 길이에 따라 128과 256으로 분류된다. |

## 핵심 360 SMURFING(스머핑)

SMURFING은 IP나 ICMP의 특성을 악용하여 엄청난 양의 데이터를 한 사이트에 집중적으로 보냄으로써 네트워크를 불능 상태로 만드는 공격 방법이다.

- 공격자는 송신 주소를 공격 대상지의 IP 주소로 위장하고 해당 네트워크 라우터의 브로드캐스트 주소를 수신지로 하여 패킷을 전송하면, 라우터의 브로드캐스트 주소로 수신된 패킷은 해당 네트워크 내의 모든 컴퓨터로 전송된다.
- 해당 네트워크 내의 모든 컴퓨터는 수신된 패킷에 대한 응답 메시지를 송신 주소인 공격 대상지로 집중적으로 전송하게 되는데, 이로 인해 공격 대상지는 네트워크 과부하로 인해 정상적인 서비스를 수행할 수 없게 된다.
- SMURFING 공격을 무력화하는 방법 중 하나는 각 네트워크 라우터에서 브로드캐스트 주소를 사용할 수 없게 미리 설정해 놓는 것이다.

## 핵심 361 DDoS(Distributed Denial of Service, 분산 서비스 거부) 공격

DDoS 공격은 여러 곳에 분산된 공격 지점에서 한 곳의 서버에 대해 분산 서비스 공격을 수행하는 것으로, 네트워크에서 취약점이 있는 호스트들을 탐색한 후 이들 호스트들에 분산 서비스 공격용 툴을 설치하여 에이전트(Agent)로 만든 후 DDoS 공격에 이용한다.

- 공격의 범위를 확대하기 위해 일부 호스트에 다수의 에이전트를 관리할 수 있는 핸들러(Handler) 프로그램을 설치하여 마스터(Master)로 지정한 후 공격에 이용하기도 한다.

### 분산 서비스 공격용 툴

에이전트(Agent)의 역할을 수행하도록 설계된 프로그램으로 데몬(Daemon)이라고 부르며, 다음과 같은 종류가 있다.

- Trin00 : 가장 초기 형태의 데몬으로, 주로 UDP Flooding 공격을 수행함
- TFN(Tribe Flooding Network) : UDP Flooding 뿐만 아니라 TCP SYN Flood 공격, ICMP 응답 요청, 스머핑 공격 등을 수행함
- TFN2K : TFN의 확장판
- Stacheldraht : 이전 툴들의 기능을 유지하면서, 공격자, 마스터, 에이전트가 쉽게 노출되지 않도록 암호화된 통신을 수행하며, 툴이 자동으로 업데이트되도록 설계됨

## 핵심 362 정보 보안 침해 공격 관련 용어

| 용어 | 의미 |
|---|---|
| 좀비(Zombie) PC | 악성코드에 감염되어 다른 프로그램이나 컴퓨터를 조종하도록 만들어진 컴퓨터로, C&C(Command & Control) 서버의 제어를 받아 주로 DDoS 공격 등에 이용된다. |
| C&C 서버 | 해커가 원격지에서 감염된 좀비 PC에 명령을 내리고 악성코드를 제어하기 위한 용도로 사용하는 서버를 말한다. |
| 봇넷(Botnet) | 악성 프로그램에 감염되어 악의적인 의도로 사용될 수 있는 다수의 컴퓨터들이 네트워크로 연결된 형태를 말한다. |
| 웜(Worm) | 네트워크를 통해 연속적으로 자신을 복제하여 시스템의 부하를 높임으로써 결국 시스템을 다운시키는 바이러스의 일종으로, 분산 서비스 거부 공격, 버퍼 오버플로 공격, 슬래머 등이 웜 공격의 한 형태이다. |
| 제로 데이 공격 (Zero Day Attack) | 보안 취약점이 발견되었을 때 발견된 취약점의 존재 자체가 널리 공표되기도 전에 해당 취약점을 통하여 이루어지는 보안 공격으로, 공격의 신속성을 의미한다. |
| 키로거 공격 (Key Logger Attack) | 컴퓨터 사용자의 키보드 움직임을 탐지해 ID, 패스워드, 계좌번호, 카드번호 등과 같은 개인의 중요한 정보를 몰래 빼가는 해킹 공격 |
| 랜섬웨어 (Ransomware) | 인터넷 사용자의 컴퓨터에 잠입해 내부 문서나 파일 등을 암호화해 사용자가 열지 못하게 하는 프로그램으로, 암호 해독용 프로그램의 전달을 조건으로 사용자에게 돈을 요구하기도 한다. |
| 백도어 (Back Door, Trap Door) | • 시스템 설계자가 서비스 기술자나 유지 보수 프로그램 작성자(Programmer)의 액세스 편의를 위해 시스템 보안을 제거하여 만들어놓은 비밀 통로로, 컴퓨터 범죄에 악용되기도 한다.<br>• 백도어 탐지 방법 : 무결성 검사, 열린 포트 확인, 로그 분석, SetUID 파일 검사 등 |
| 트로이 목마 (Trojan Horse) | 정상적인 기능을 하는 프로그램으로 위장하여 프로그램 내에 숨어 있다가 해당 프로그램이 동작할 때 활성화되어 부작용을 일으키는 것으로, 자기 복제 능력은 없다. |

## 핵심 363 보안 서버의 개념

보안 서버란 인터넷을 통해 개인정보를 암호화하여 송 · 수신할 수 있는 기능을 갖춘 서버를 말한다.

- '개인정보의 기술적 · 관리적 보호조치 기준'에 따르면 보안 서버는 다음과 같은 기능을 갖춰야 한다.
  - 서버에 SSL(Secure Socket Layer) 인증서를 설치하여 전송 정보를 암호화하여 송 · 수신하는 기능
  - 서버에 암호화 응용 프로그램을 설치하고 전송 정보를 암호화하여 송 · 수신하는 기능
- 스니핑(Sniffing)을 이용한 정보 유출, 피싱(Phishing)을 이용한 위조 사이트 등에 대비하기 위해 보안 서버 구축이 필요하다.

## 핵심 364 인증(認證, Authentication)의 개념

인증은 다중 사용자 컴퓨터 시스템이나 네트워크 시스템에서 로그인을 요청한 사용자의 정보를 확인하고 접근 권한을 검증하는 보안 절차이다.

- 인증에는 네트워크를 통해 컴퓨터에 접속하는 사용자의 등록 여부를 확인하는 것과 전송된 메시지의 위 · 변조 여부를 확인하는 것이 있다.
- 인증의 주요 유형
  - 지식 기반 인증(Something You Know) : 사용자가 기억하고 있는 정보를 기반으로 인증을 수행하는 것
  - 소유 기반 인증(Something You Have) : 사용자가 소유하고 있는 것을 기반으로 인증을 수행하는 것
  - 생체 기반 인증(Something You Are) : 사용자의 고유한 생체 정보를 기반으로 인증을 수행하는 것
  - 위치 기반 인증(Somewhere You Are) : 인증을 시도하는 위치의 적절성 확인하는 것

## 핵심 365 보안 아키텍처(Security Architecture)

보안 아키텍처란 정보 시스템의 무결성(Integrity), 가용성(Availability), 기밀성(Confidentiality)을 확보하기 위해 보안 요소 및 보안 체계를 식별하고 이들 간의 관계를 정의한 구조를 말한다.

- 보안 아키텍처를 통해 관리적, 물리적, 기술적 보안 개념의 수립, 보안 관리 능력의 향상, 일관된 보안 수준의 유지를 기대할 수 있다.
- 보안 아키텍처는 보안 수준에 변화가 생겨도 기본 보안 아키텍처의 수정 없이 지원할 수 있어야 한다.
- 보안 아키텍처는 보안 요구사항의 변화나 추가를 수용할 수 있어야 한다.

## 핵심 366 보안 프레임워크(Security Framework)

프레임워크는 '뼈대', '골조'를 의미하는 용어이며, 보안 프레임워크는 안전한 정보 시스템 환경을 유지하고 보안 수준을 향상시키기 위한 체계를 말한다.

- ISO 27001은 정보보안 관리를 위한 국제 표준으로, 일종의 보안 인증이자 가장 대표적인 보안 프레임워크이다.
- ISO 27001은 영국의 BSI(British Standards Institute)가 제정한 BS 7799를 기반으로 구성되어 있다.
- ISO 27001은 조직에 대한 정보보안 관리 규격이 정의되어 있어 실제 심사/인증용으로 사용된다.

## 핵심 367 리눅스(LINUX) 로그

리눅스에서는 시스템의 모든 로그를 var/log 디렉터리에서 기록하고 관리한다.

- 로그 파일을 관리하는 syslogd 데몬은 etc/syslog.conf 파일을 읽어 로그 관련 파일들의 위치를 파악한 후 로그 작업을 시작한다.
- syslog.conf 파일을 수정하여 로그 관련 파일들의 저장 위치와 파일명을 변경할 수 있다.

## 핵심 368 리눅스의 주요 로그 파일

| 로그 | 파일명 | 데몬 | 내용 |
|---|---|---|---|
| 커널 로그 | /dev/console | kernel | 커널에 관련된 내용을 관리자에게 알리기 위해 파일로 저장하지 않고 지정된 장치에 표시한다. |
| 부팅 로그 | /var/log/boot.log | boot | 부팅 시 나타나는 메시지들을 기록한다. |
| 크론 로그 | /var/log/cron | crond | 작업 스케줄러인 crond의 작업 내역을 기록한다. |
| 시스템 로그 | /var/log/messages | syslogd | 커널(kernel)에서 실시간으로 보내오는 메시지들을 기록한다. |
| 보안 로그 | /var/log/secure | xinetd | 시스템의 접속에 대한 로그를 기록한다. |
| FTP 로그 | /var/log/xferlog | ftpd | FTP로 접속하는 사용자에 대한 로그를 기록한다. |
| 메일 로그 | /var/log/maillog | sendmail popper | 송수신 메일에 대한 로그를 기록한다. |

## 핵심 369 Windows 이벤트 뷰어의 로그

| 로그 | 내용 |
|---|---|
| 응용 프로그램 | • 응용 프로그램에서 발생하는 이벤트가 기록된다.<br>• 기록되는 이벤트는 응용 프로그램 개발자에 의해 결정된다. |
| 보안 | 로그온 시도, 파일이나 객체 생성, 조회, 제거 등의 리소스 사용과 관련된 이벤트가 기록된다. |
| 시스템 | Windows 시스템 구성 요소에 의해 발생하는 이벤트가 기록된다. |
| Setup | 프로그램 설치와 관련된 이벤트가 기록된다. |
| Forwarded Events | 다른 컴퓨터와의 상호 작용으로 발생하는 이벤트가 기록된다. |

## 핵심 370 보안 솔루션

### 보안 솔루션의 개념

보안 솔루션이란 접근 통제, 침입 차단 및 탐지 등을 수행하여 외부로부터의 불법적인 침입을 막는 기술 및 시스템을 말한다.

- 주요 보안 솔루션에는 방화벽, 침입 탐지 시스템(IDS), 침입 방지 시스템(IPS), 데이터 유출 방지(DLP), 웹 방화벽, VPN, NAC 등이 있다.

| | |
|---|---|
| **방화벽(Firewall)** | 기업이나 조직 내부의 네트워크와 인터넷 간에 전송되는 정보를 선별하여 수용 · 거부 · 수정하는 기능을 가진 침입 차단 시스템이다. |
| **침입 탐지 시스템(IDS; Intrusion Detection System)** | 컴퓨터 시스템의 비정상적인 사용, 오용, 남용 등을 실시간으로 탐지하는 시스템이다. |
| **침입 방지 시스템(IPS; Intrusion Prevention System)** | • 방화벽과 침입 탐지 시스템을 결합한 것이다.<br>• 비정상적인 트래픽을 능동적으로 차단하고 격리하는 등의 방어 조치를 취하는 보안 솔루션이다. |
| **데이터 유출 방지(DLP; Data Leakage/Loss Prevention)** | 데이터 유출 방지는 내부 정보의 외부 유출을 방지하는 보안 솔루션이다. |
| **웹 방화벽(Web Firewall)** | 일반 방화벽이 탐지하지 못하는 SQL 삽입 공격, Cross-Site Scripting(XSS) 등의 웹 기반 공격을 방어할 목적으로 만들어진 웹 서버에 특화된 방화벽이다. |
| **VPN(Virtual Private Network, 가상 사설 통신망)** | 가상 사설 네트워크로서 인터넷 등 통신 사업자의 공중 네트워크와 암호화 기술을 이용하여 사용자가 마치 자신의 전용 회선을 사용하는 것처럼 해주는 보안 솔루션이다. |
| **NAC(Network Access Control)** | 네트워크에 접속하는 내부 PC의 MAC 주소를 IP 관리 시스템에 등록한 후 일관된 보안 관리 기능을 제공하는 보안 솔루션이다. |
| **ESM(Enterprise Security Management)** | 다양한 장비에서 발생하는 로그 및 보안 이벤트를 통합하여 관리하는 보안 솔루션이다. |